PROJECT REPORT

Computer System and Microprocessor

Final Project Report

Dryer Control System Using the 8086 Microprocessor

STUDENT NAME	**Asem Ameen Qaid**
NIM	**24524038**
CLASS	**IP**
DATE	**22 January 2026**

Laboratory of Computer Systems and Microprocessor

Undergraduate Program of Electrical Engineering

Faculty of Industrial Technology

Islamic University of Indonesia

2026

ABSTRACT

This report presents the design and simulation of a microprocessor-based dryer control system using the Intel 8086. The system monitors temperature and pressure and controls a heater and an alarm according to user-defined setpoints. An LM35 temperature sensor and an MPX4115A pressure sensor produce analog signals that are converted by an ADC0804 into 8-bit digital values. An 8255 Programmable Peripheral Interface, configured in Mode 0, handles all input and output: a two-digit seven-segment display, the ADC data, four push buttons, the heater relay and the buzzer.

The control program, written in 8086 assembly language, implements on–off temperature control and a pressure alarm, with start/stop operation and adjustable setpoints. The complete system was designed and simulated in Proteus. Fifteen test cases at setpoints of 30 °C and 100 kPa were simulated, and in every case the heater and buzzer responded exactly as specified by the control logic.

Keywords: 8086 microprocessor, 8255 PPI, ADC0804, assembly language, temperature control, pressure alarm, Proteus simulation

CONTENTS

1 Introduction

2 Theoretical Basis

3 System Design

4 Software Design

5 Simulation Results

6 Analysis and Discussion

7 Conclusion

1 Introduction

Drying processes require the temperature inside the dryer to be held near a target value while unsafe operating conditions are detected and reported. This project implements such a controller with the Intel 8086 microprocessor. The system measures temperature and pressure, switches a heater to regulate temperature, and activates an alarm when pressure exceeds a safe limit.

The objectives of the project are:

- to acquire temperature and pressure through analog sensors and an analog-to-digital converter;
- to interface the sensors, display, push buttons and actuators with the 8086 through an 8255 PPI;
- to implement on–off heater control and an over-pressure alarm in 8086 assembly language;
- to allow the operator to start and stop the system and adjust both setpoints;
- to verify correct operation under different conditions by simulation in Proteus.

2 Theoretical Basis

2.1 8086 Microprocessor

The 8086 is a 16-bit processor that serves as the main control unit of the system. It performs all decision-making operations: reading sensor data, comparing values with setpoints and generating control signals for the output devices. It communicates with memory and peripherals through its data and address buses and executes the control program stored in memory.

2.2 8255 Programmable Peripheral Interface

The 8255 PPI connects the 8086 to external devices. It provides three programmable 8-bit ports (A, B and C), and Port C can be split into two independent 4-bit halves. Each port can be configured as input or output by writing a control word to its control register.

2.3 ADC0804

The ADC0804 is an 8-bit analog-to-digital converter. It converts the analog output voltages of the LM35 and MPX4115A sensors into digital values that the microprocessor can process, using a start-conversion and end-of-conversion mechanism.

2.4 LM35 Temperature Sensor

The LM35 produces an output voltage linearly proportional to temperature in degrees Celsius. It is used to monitor the temperature inside the dryer.

2.5 MPX4115A Pressure Sensor

The MPX4115A produces an analog output voltage proportional to the applied pressure. After conversion, the value is compared with a pressure limit to activate the alarm when necessary.

2.6 Output Devices

- **Relay and heater:** the relay switches the heater, which is turned on while the temperature is below the setpoint. A lamp indicates heater operation.
- **Buzzer:** an audible alarm activated when the pressure exceeds the allowable limit.
- **Seven-segment display:** shows the temperature or pressure value, depending on the selected mode.

3 System Design

3.1 Block Diagram

Figure 1 shows the overall structure. The 8086 communicates with the 8255 over the system bus. Port A drives the display, Port B reads the ADC, the lower half of Port C controls the ADC conversions and the actuators, and the upper half of Port C reads the push buttons.

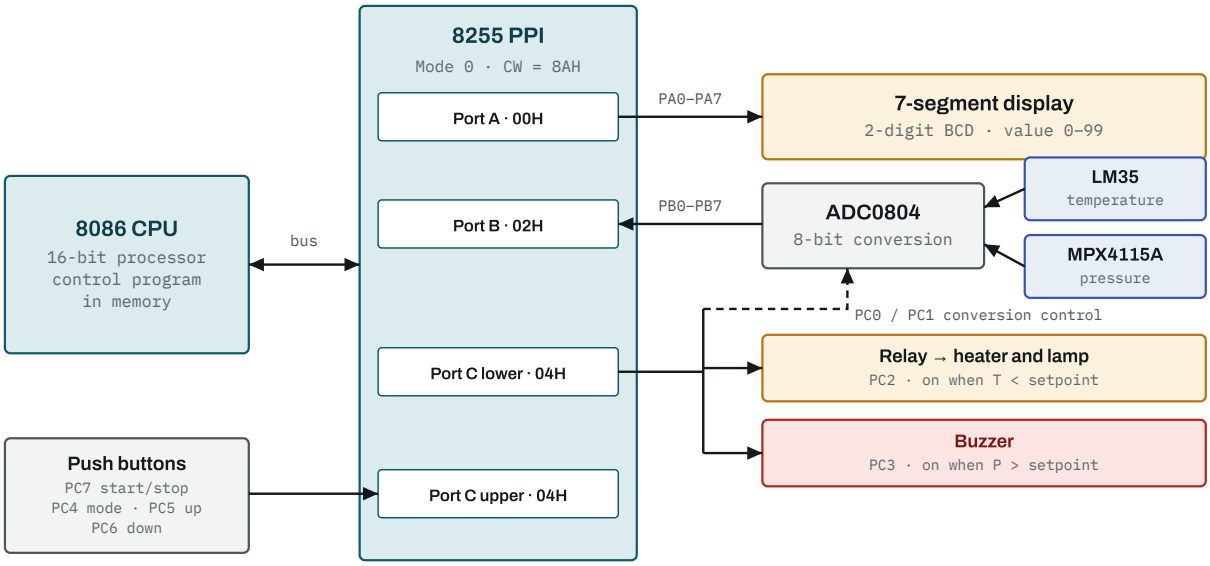


Figure 1. System block diagram of the 8086 dryer controller.

3.2 I/O Configuration

The 8255 is initialised with the control word 8AH (Figure 2), which places both groups in Mode 0, makes Port A an output, Port B an input, the upper half of Port C an input and the lower half of Port C an output. Table 1 lists the complete I/O map used by the program.

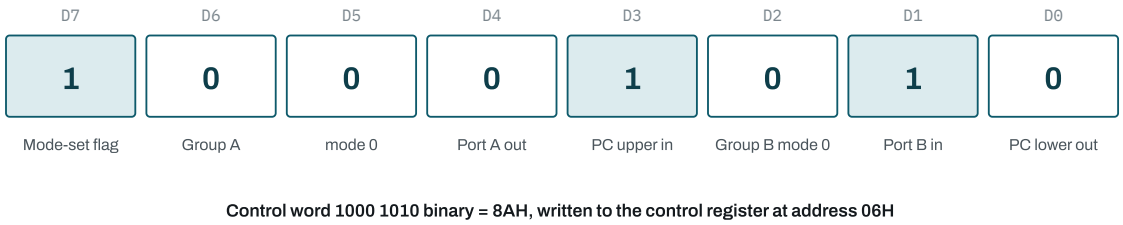


Figure 2. Bit pattern of the 8255 control word.

Table 1. I/O map

Port / bit	Address	Direction	Function
`Port A`	`00H`	Output	Two-digit BCD value to the seven-segment display
`Port B`	`02H`	Input	8-bit data from the ADC0804
`PC0`	`04H`	Output	Temperature conversion control (active low)
`PC1`	`04H`	Output	Pressure conversion control (active low)
`PC2`	`04H`	Output	Heater relay and lamp
`PC3`	`04H`	Output	Buzzer (pressure alarm)
`PC4`	`04H`	Input	MODE button: temperature or pressure (active low)
`PC5 / PC6`	`04H`	Input	UP / DOWN buttons: adjust setpoint (active low)
`PC7`	`04H`	Input	START/STOP button (active low)
`Control`	`06H`	Output	8255 control register (8AH)

3.3 Simulation Circuit

The complete circuit was designed and simulated in Proteus (Figure 3). It includes the 8086, the 8255, the analog-to-digital conversion stage with both sensors, the push buttons, the seven-segment display, the relay with heater and lamp, and the buzzer.

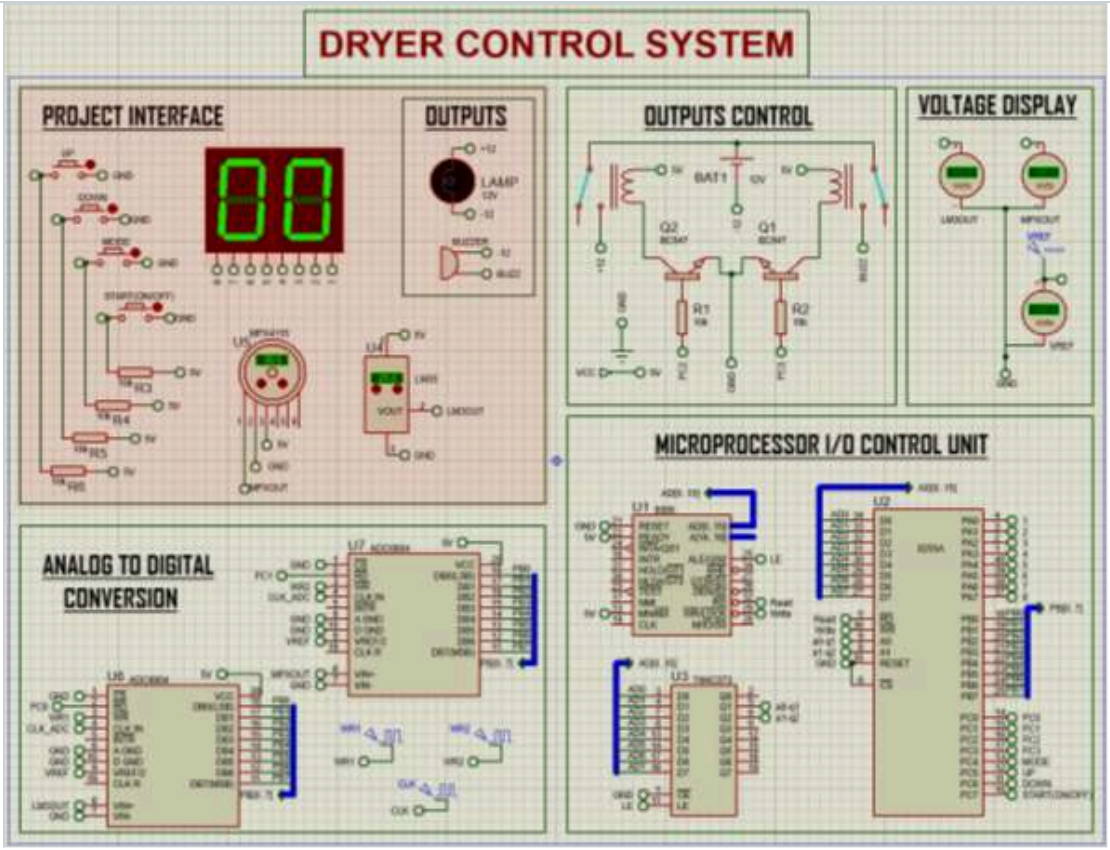


Figure 3. Proteus simulation circuit of the dryer control system.

4 Software Design

4.1 Program Structure

After initialising the data segment and the 8255, the program enters a continuous main loop (Figure 4). In each pass it reads the buttons on the upper half of Port C. START/STOP toggles between the RUN and STOP states, MODE switches the display and adjustment between temperature and pressure, and UP/DOWN change the selected setpoint while the system is stopped. Each button press is debounced with a software delay, and the program waits for the button to be released before continuing.

In the STOP state the heater and buzzer are switched off and the selected setpoint is shown on the display. In the RUN state the program reads both sensors, applies the control logic and displays the measured value.

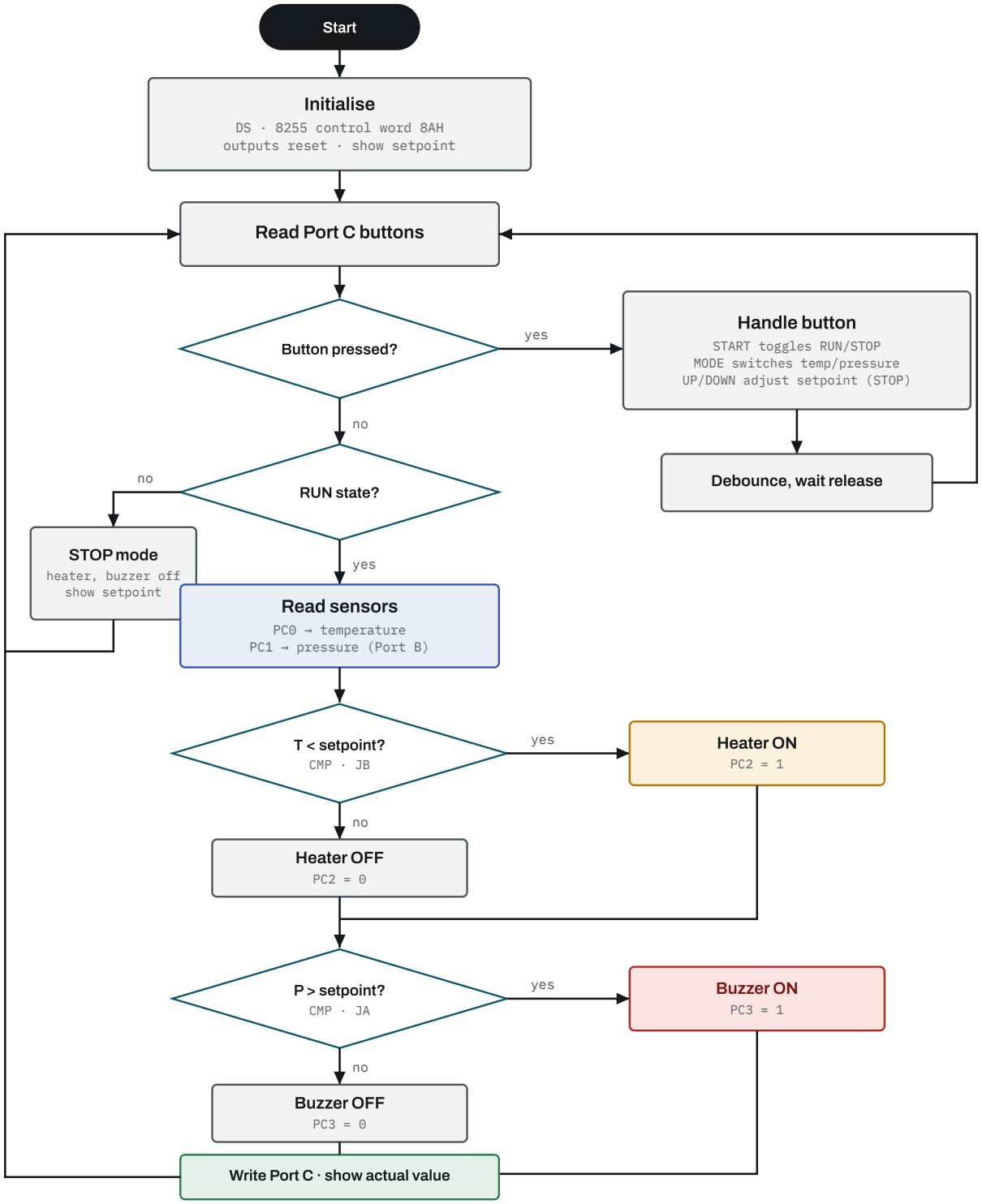


Figure 4. Program flowchart of the dryer controller.

4.2 Sensor Acquisition

Each sensor is read by pulling its conversion control line low (PC0 for temperature, PC1 for pressure), waiting a short software delay for the conversion, reading the result from Port B and returning the control line high. The values are stored in `TEMP_ACTUAL` and `PRESS_ACTUAL`.

4.3 Control Logic

Temperature control uses an unsigned comparison: the heater is switched on while the measured temperature is below the setpoint (`CMP` and `JB`) and switched off otherwise. The pressure alarm is switched on while the measured pressure is above its setpoint (`CMP` and `JA`).

Because the buttons and the outputs share Port C, the program never writes outputs directly. It keeps a copy of the output state in the variable `OUT_IMAGE`, changes individual bits with `AND` and `OR` masks, and writes only the lower four bits to the port. Writing an output therefore never disturbs the button inputs.

```
; ---- Heater control: ON while temperature is below setpoint ----
MOV  AL, TEMP_ACTUAL
CMP  AL, TEMP_SETPOINT
JB   TURN_ON_HEATER
AND  OUT_IMAGE, 11111011B      ; PC2 = 0, heater OFF
JMP  CHECK_ALARM
TURN_ON_HEATER:
OR   OUT_IMAGE, 00000100B      ; PC2 = 1, heater ON
CHECK_ALARM:
MOV  AL, PRESS_ACTUAL
CMP  AL, PRESS_SETPOINT
JA   TURN_ON_BUZZER
AND  OUT_IMAGE, 11110111B      ; PC3 = 0, buzzer OFF
JMP  UPDATE_OUTPUT
TURN_ON_BUZZER:
OR   OUT_IMAGE, 00001000B      ; PC3 = 1, buzzer ON
UPDATE_OUTPUT:
CALL SEND_OUTPUT               ; write lower nibble to Port C
CALL DISPLAY_ACTUAL            ; BCD value to Port A
```

4.4 Display

The value to be displayed is converted to two-digit BCD by dividing by 10: the quotient forms the tens digit in the upper four bits and the remainder forms the units digit in the lower four bits. The result is written to Port A, which drives the seven-segment display. The display therefore shows values from 0 to 99.

5 Simulation Results

To evaluate the system, a series of simulation tests was carried out in Proteus. The input values of the LM35 and MPX4115A were varied to represent different operating conditions, with the temperature setpoint at 30 °C and the pressure setpoint at 100 kPa. For each case the states of the heater (relay) and the buzzer (alarm) were observed and recorded (Table 2). The last column compares each observed result with the behaviour required by the control logic (heater on when T < 30 °C; buzzer on when P > 100 kPa).

Table 2. Simulation results (setpoints: 30 °C, 100 kPa)

No.	Temperature (°C)	Pressure (kPa)	Heater	Buzzer	As specified
1	22	85	ON	OFF	Yes
2	25	95	ON	OFF	Yes
3	28	98	ON	OFF	Yes
4	29	102	ON	ON	Yes
5	30	90	OFF	OFF	Yes
6	30	105	OFF	ON	Yes
7	32	88	OFF	OFF	Yes
8	33	110	OFF	ON	Yes
9	35	97	OFF	OFF	Yes
10	36	115	OFF	ON	Yes
11	27	92	ON	OFF	Yes
12	26	108	ON	ON	Yes
13	31	99	OFF	OFF	Yes
14	34	101	OFF	ON	Yes
15	24	120	ON	ON	Yes

6 Analysis and Discussion

All fifteen cases produced the expected output. The heater and lamp were activated whenever the measured temperature was below the setpoint, allowing the dryer temperature to rise toward the desired level, and were deactivated once the temperature reached or exceeded the setpoint. This on–off behaviour prevents overheating and confirms that the comparison implemented with `CMP` and `JB` works as intended. Cases 5 and 6, at exactly 30 °C, confirm that the heater switches off at the setpoint itself.

The pressure protection behaved correctly in every case: the buzzer activated whenever the pressure exceeded 100 kPa and switched off automatically when the pressure returned to the safe range, without manual intervention. Cases 13 and 14 (99 and 101 kPa) show the alarm boundary.

Because the program runs in a continuous loop, sensor readings and output decisions are updated in real time. Separating inputs and outputs within Port C through the `OUT_IMAGE` masking technique avoided input/output conflicts, so writing outputs never interfered with the button readings.

The system has the expected limitations of a simple design. On–off control without hysteresis can cause frequent heater switching near the setpoint, and the two-digit display limits shown values to 0–99. The design was verified in simulation.

7 Conclusion

A dryer control system based on the 8086 microprocessor was designed and simulated. The system monitors temperature and pressure with appropriate sensors and converts the analog signals into digital data with the ADC0804. The 8255 PPI provides efficient input and output interfacing, and the control program operates the heater and the alarm correctly. The simulation results show that the system meets its requirements under all tested conditions.

The design can be extended by adding an LCD display, hysteresis or more advanced control techniques, and further safety thresholds or sensors, largely through software changes without significant changes to the hardware structure.