2507/302
MICROCONTROLLER TECHNOLOGY
June/July 2019
Time: 3 hours

THE KENYA NATIONAL EXAMINATIONS COUNCIL

DIPLOMA IN AERONAUTICAL ENGINEERING (AVIONICS OPTION)

MODULE III

MICROCONTROLLER TECHNOLOGY

3 hours

INSTRUCTIONS TO CANDIDATES

You should have the following for this examination.
Answer booklet;
Non-programmable scientific calculator;
8051 instructions set.
*This paper consists of **EIGHT** questions.*
*Answer any **FIVE** of the **EIGHT** questions in the answer booklet provided.*
All questions carry equal marks.
Maximum marks for each part of a question are as indicated.
Candidates should answer the questions in English.

This paper consists of 10 printed pages.

Candidates should check the question paper to ascertain that all the pages are printed as indicated and that no questions are missing.

Turn over

1. (a) State **two** functions of each of the following microcontroller parts:

(i) ALU;

(ii) Bus system;

(iii) EEPROM. → No of addresses are limited.

(6 marks)

(b) Convert:

(i) 78.18175_{10} into binary; ÷2

(ii) 430.03125_{10} into hexadecimal.

(6 marks)

(c) Work out $10110_2 \times 1101_2$ and give your answer in octal. (4 marks)

20

(d) Work out 1111_2 - 10000_2 using 8-bits Two's complement method. (4 marks)

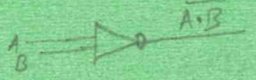
21-17=4 4=(100)

2. (a) Explain how the 8051 microcontroller addresses 16-bit external memory. (3 marks)

(b) (i) Draw ladder rung symbols for each of the following:

(I) normally closed contact;

(II) NAND gate.



(3 marks)

(ii) Table 1 shows a PLC program listing.

Table 1

LD X405
OR X401
AN 1 X402
OUT Y430
END

open

Closed

16

Draw its ladder diagram. (4 marks)

(c) Describe each of the following PLC fault finding techniques:

(i) Replication;

(ii) Timing-checks;

(iii) Last-output-set.

(6 marks)

(d) Explain each of the following 8051 microcontroller instructions:

(i) DRL A, Rn;

(ii) RET.

(4 marks)

3. (a) State:

(i) **two** types of stepper motors;

(ii) **three** applications of stepper motors in robotics.

(5 marks)

(b) Table 2 represents a set of Intel 8051 instructions. Complete the table by filling in the missing fields. (8 marks)

Instruction		Hexadecimal code	Addressing mode
(i)	MOV A, @ R1 Registing mode		
(ii)	MOV A, # 65H		
(iii)	ADD A, 58H		
(iv)	MOVCA, @ A + DPTR		

(c) Figure 1 shows a block diagram of a microcontroller based temperature controlled system.

R1

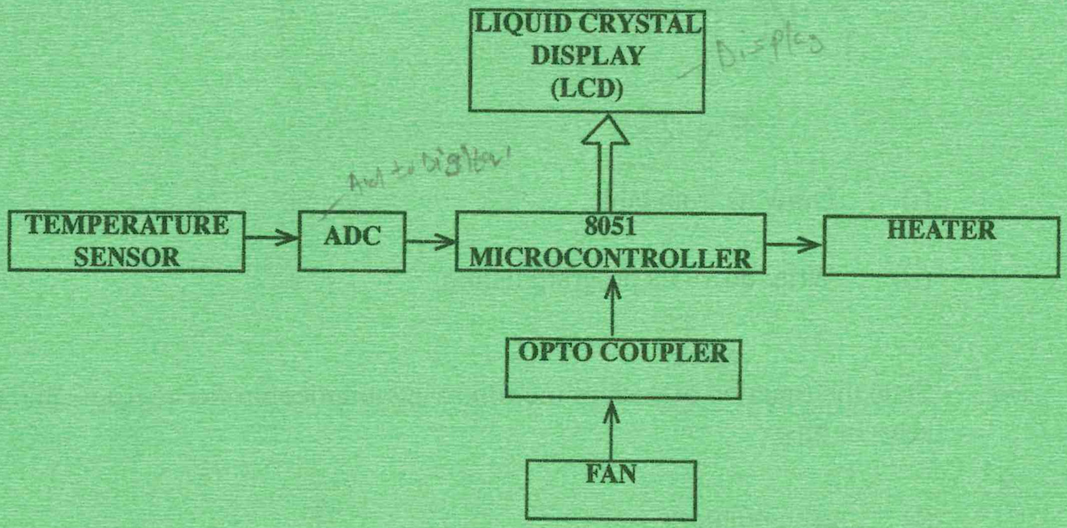


Fig. 1

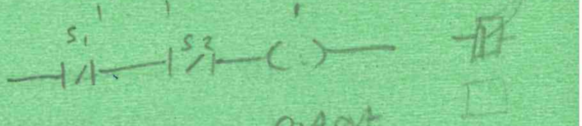
With the aid of a flow chart, explain how the system can be used to regulate the temperature to within 20°C and 26°C. (7 marks)

4. (a) With the aid of a one line instruction, describe **three** microcontroller instruction groups. (9 marks)

(b) State **three** advantages of using PLCs over computers for control tasks. (3 marks)

faster

(c) A motor is controlled by three switches. For a proper motor operation either of two normally open switches; S_1 and S_2, have to be closed and a normally closed switch, S_3 have to be opened. For the controller, draw:



(i) the truth table representation;

(ii) functional block diagram;

(Iii) ladder rung diagram. (8 marks)

S1	S2	S3	output
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

5. (a) Explain the basic steps a microcontroller undergoes on receiving an interrupt request. (5 marks)

(b) With the aid of a flow chart, write a microcontroller program to generate a square wave, with 50% duty cycle on port PO whose address is 07H. (8 marks)

(c) Figure 2 shows a PLC ladder diagram:

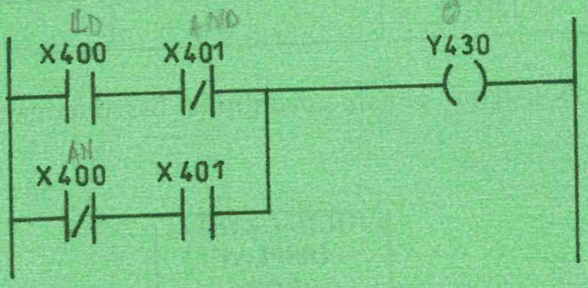


Fig. 2

(i) Draw its truth table.

(ii) Write down its program listing. (7 marks)

12/20

6. (a) State the function of each of the following microcontroller registers and their typical sizes in bits:

(i) accumulator;

(ii) program status word (PSW);

(iii) data pointer. (6 marks)

(b) Figure 3 shows the I/O port structure of a microcontroller.

(MSB) (LSB)

$\overline{RD}$	$\overline{WR}$	T_1	$\overline{T_0}$	$\overline{INT_1}$	$\overline{INT_0}$	TXD	RXD

Fig. 3

Describe the function of the following:

(i) T1;

(ii) TXD;

(iii) RXD.

(6 marks)

(c) Write down an assembly language program to add two 16-bit numbers 80 AOH and 2070 H and store the result in the memory location 5 OH and 51 H. (8 marks)

7. (a) (i) State **two** uses of RAM in a microcontroller.

(ii) Figure 4 represents timing waveforms for a microcontroller memory read cycle. Explain the memory read process. (8 marks)

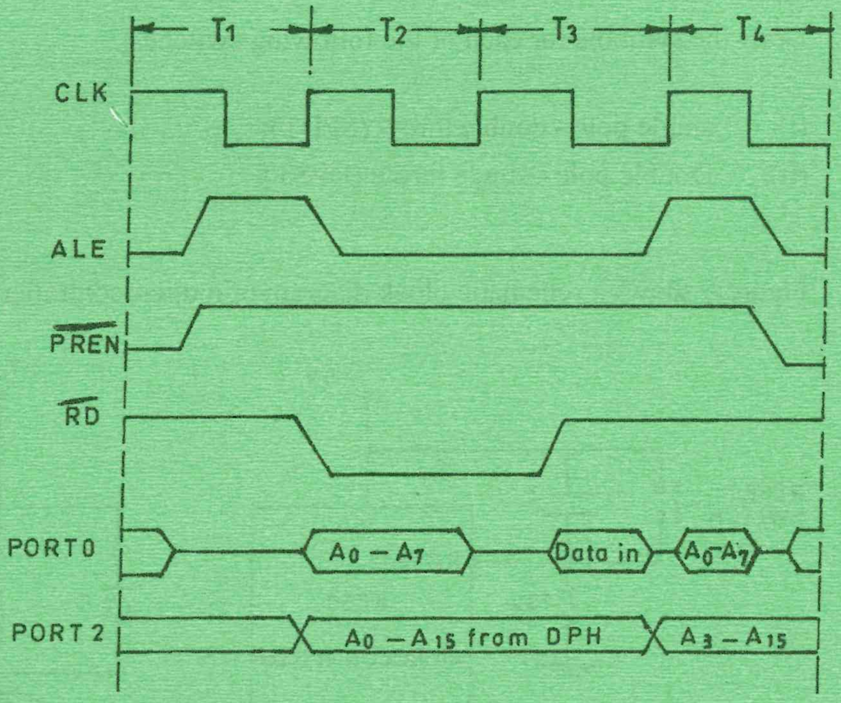


Fig. 4

(b) Figure 5 shows a schematic diagram of an electronic component.

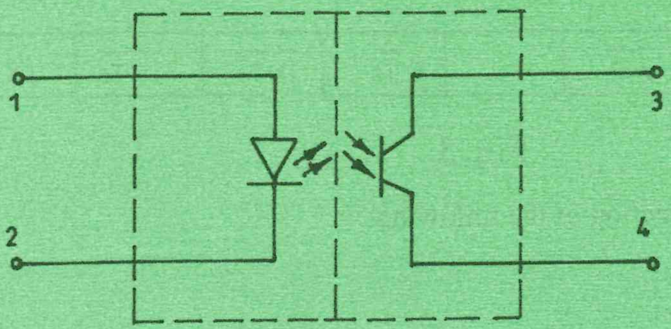


Fig. 5

For the circuit:

(i) identify the component;

(ii) describe its operation;

(iii) state **two** application areas.

(6 marks)

(c) Describe **three** functions of a Human Machine Interface (HMI) in an international airport flight control system. (6 marks)

8. (a) Draw the symbols for each of the following switches:

(i) single pole - double throw (SPDT);

(ii) Double pole - single throw (DPST).

(4 marks)

(b) Figure 6 shows a schematic block diagram of a microcontroller-based system.

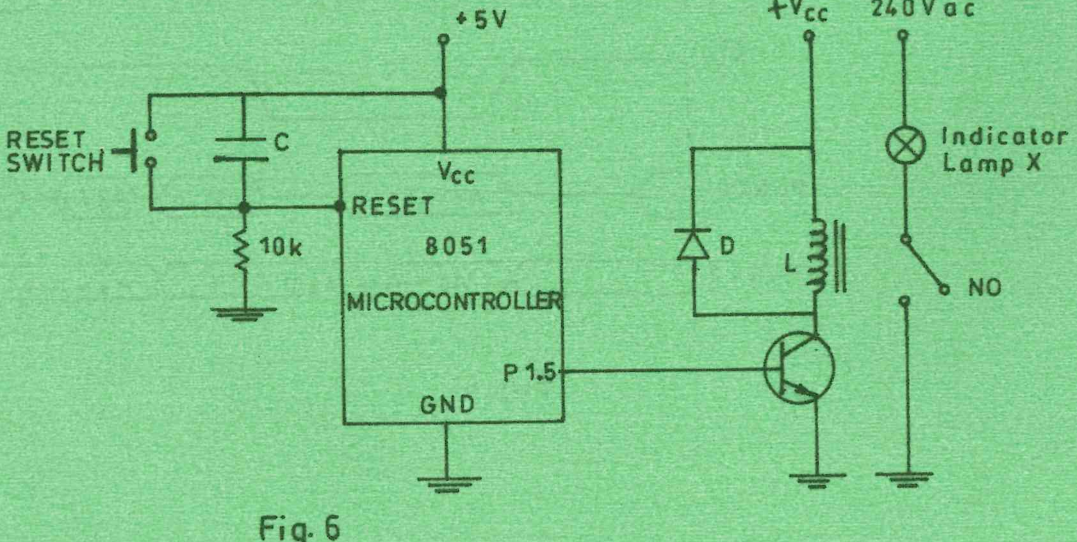


Fig. 6

(i) Explain the circuit operation.

(ii) Write an assembly language program to:

configure port, P1.5 as output;
set lamp X on;
lamp remains on for a predetermined delay, DELAY;
switch off lamp X;
repeat the process.

(10 marks)

(c) Table 3 shows a microcontroller program listing.

LABEL	PROGRAM INSTRUCTION
	MOV A, #OOH
	MOV RO, #OA H
LOOP:	DEC RO
	DEC RO
	ADD A, RO
	JNZ LOOP

With the aid of a trace table, determine the contents of register A at the end of the program execution.

(6 marks)

	A	RO
MOVA H 00H	00	–
MOVRO #OAH	00	OA.
Dec RO		
Dec RO		
ADDA, RO	OA	OA

Hex Code	Number of Bytes	**Mnemonic**	**Operands**
00	1	**NOP**	
01	2	**AJMP**	code addr
02	3	**LJMP**	code addr
03	1	**RR**	A
04	1	**INC**	A
05	2	**INC**	data addr
06	1	**INC**	@R0
07	1	**INC**	@R1
08	1	**INC**	R0
09	1	**INC**	R1
0A	1	**INC**	R2
0B	1	**INC**	R3
0C	1	**INC**	R4
0D	1	**INC**	R5
0E	1	**INC**	R6
0F	1	**INC**	R7
10	3	**JBC**	bit addr, code addr
11	2	**ACALL**	code addr
12	3	**LCALL**	code addr
13	1	**RRC**	A
14	1	**DEC**	A
15	2	**DEC**	data addr
16	1	**DEC**	@R0
17	1	**DEC**	@R1
18	1	**DEC**	R0
19	1	**DEC**	R1
1A	1	**DEC**	R2
1B	1	**DEC**	R3
1C	1	**DEC**	R4
1D	1	**DEC**	R5
1E	1	**DEC**	R6
1F	1	**DEC**	R7
20	3	**JB**	bit addr, code addr
21	2	**AJMP**	code addr
22	1	**RET**	
23	1	**RL**	A
24	2	**ADD**	A,#data
25	2	**ADD**	A,data addr
26	1	**ADD**	A,@R0
27	1	**ADD**	A,@R1
28	1	**ADD**	A,R0
29	1	**ADD**	A,R1
2A	1	**ADD**	A,R2
2B	1	**ADD**	A,R3
2C	1	**ADD**	A,R4
2D	1	**ADD**	A,R5
2E	1	**ADD**	A,R6
2F	1	**ADD**	A,R7
30	3	**JNB**	bit addr, code addr
31	2	**ACALL**	code addr
32	1	**RETI**	
33	1	**RLC**	A
34	2	**ADDC**	A,#data
35	2	**ADDC**	A,data addr
36	1	**ADDC**	A,@R0
37	1	**ADDC**	A,@R1
38	1	**ADDC**	A,R0
39	1	**ADDC**	A,R1
3A	1	**ADDC**	A,R2
3B	1	**ADDC**	A,R3
3C	1	**ADDC**	A,R4
3D	1	**ADDC**	A,R5
3E	1	**ADDC**	A,R6
3F	1	**ADDC**	A,R7
40	2	**JC**	code addr
41	2	**AJMP**	code addr
42	2	**ORL**	data addr,A
43	3	**ORL**	data addr,#data
44	2	**ORL**	A,#data
45	2	**ORL**	A,data addr
46	1	**ORL**	A,@R0
47	1	**ORL**	A,@R1
48	1	**ORL**	A,R0
49	1	**ORL**	A,R1
4A	1	**ORL**	A,R2
4B	1	**ORL**	A,R3
4C	1	**ORL**	A,R4
4D	1	**ORL**	A,R5
4E	1	**ORL**	A,R6
4F	1	**ORL**	A,R7
50	2	**JNC**	code addr
51	2	**ACALL**	code addr
52	2	**ANL**	data addr,A
53	3	**ANL**	data addr,#data
54	2	**ANL**	A,#data
55	2	**ANL**	A,data addr
56	1	**ANL**	A,@R0
57	1	**ANL**	A,@R1
58	1	**ANL**	A,R0
59	1	**ANL**	A,R1
5A	1	**ANL**	A,R2
5B	1	**ANL**	A,R3
5C	1	**ANL**	A,R4
5D	1	**ANL**	A,R5
5E	1	**ANL**	A,R6
5F	1	**ANL**	A,R7
60	2	**JZ**	code addr
61	2	**AJMP**	code addr

8051 OpCodes en Hexadecinal.

62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr, data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1

8051 OpCodes en Hexadecinal.

C8	1	**XCH**	A,R0
C9	1	**XCH**	A,R1
CA	1	**XCH**	A,R2
CB	1	**XCH**	A,R3
CC	1	**XCH**	A,R4
CD	1	**XCH**	A,R5
CE	1	**XCH**	A,R6
CF	1	**XCH**	A,R7
D0	2	**POP**	data addr
D1	2	**ACALL**	code addr
D2	2	**SETB**	bit addr
D3	1	**SETB**	C
D4	1	**DA**	A
D5	3	**DJNZ**	data addr,code addr
D6	1	**XCHD**	A,@R0
D7	1	**XCHD**	A,@R1
D8	2	**DJNZ**	R0,code addr
D9	2	**DJNZ**	R1,code addr
DA	2	**DJNZ**	R2,code addr
DB	2	**DJNZ**	R3,code addr
DC	2	**DJNZ**	R4,code addr
DD	2	**DJNZ**	R5,code addr
DE	2	**DJNZ**	R6,code addr
DF	2	**DJNZ**	R7,code addr
E0	1	**MOVX**	A,@DPTR
E1	2	**AJMP**	code addr
E2	1	**MOVX**	A,@R0
E3	1	**MOVX**	A,@R1
E4	1	**CLR**	A
E5	2	**MOV**	A,data addr
E6	1	**MOV**	A,@R0
E7	1	**MOV**	A,@R1
E8	1	**MOV**	A,R0
E9	1	**MOV**	A,R1
EA	1	**MOV**	A,R2
EB	1	**MOV**	A,R3
EC	1	**MOV**	A,R4
ED	1	**MOV**	A,R5
EE	1	**MOV**	A,R6
EF	1	**MOV**	A,R7
F0	1	**MOVX**	@DPTR,A
F1	2	**ACALL**	code addr
F2	1	**MOVX**	@R0,A
F3	1	**MOVX**	@R1,A
F4	1	**CPL**	A
F5	2	**MOV**	data addr,A
F6	1	**MOV**	@R0,A
F7	1	**MOV**	@R1,A
F8	1	**MOV**	R0,A
F9	1	**MOV**	R1,A
FA	1	**MOV**	R2,A
FB	1	**MOV**	R3,A
FC	1	**MOV**	R4,A
FD	1	**MOV**	R5,A
FE	1	**MOV**	R6,A
FF	1	**MOV**	R7,A

Instruction Opcodes in Hexadecimal Order

8051 OpCodes en Hexadecinal.

THIS IS THE LAST PRINTED PAGE.