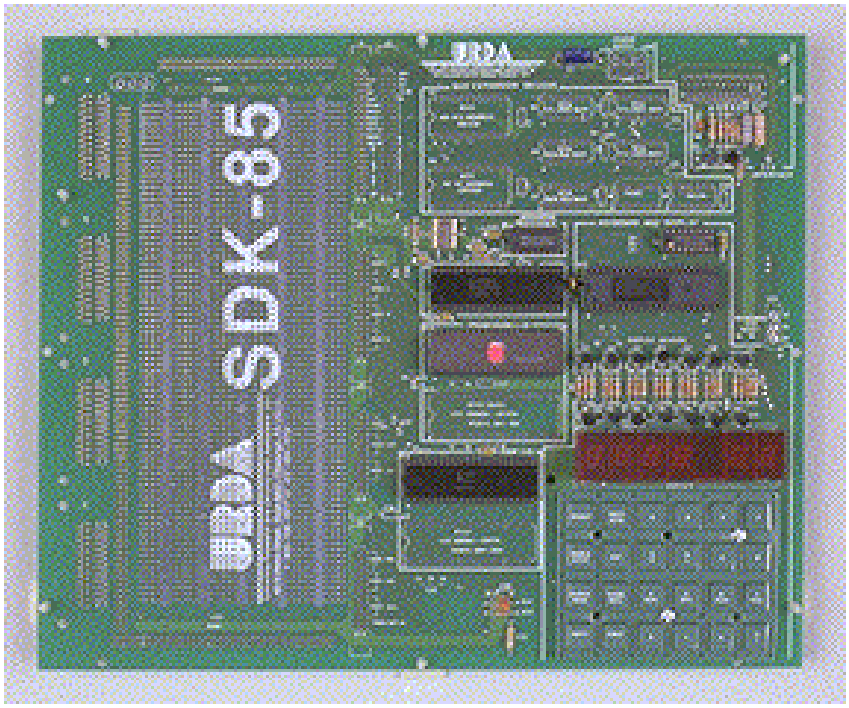
NEW YORK CITY TECHNICAL COLLEGE
City University of New York

DIGITAL ELECTRONICS II



Digital-to-Analog Converter interfaced with the SDK-85

ET 482 final project

ET 482	
Authors:	Nedzad Joldic
	Artur Brozyna
Instructor:	Prof. A. Roitstein
Date:	December 21, 2001

TABLE OF CONTENTS

Objective

This project’s objective is to design hardware interface and write the assembly language program that will be connected to and be able to communicate with the SDK-85. Switch is to be used for the input of data to the SDK-85. LED’s and Address/Data Fields of SKD-85 are to be used for the output of information.

Project Analysis

This project utilizes I/O functionalities of the Intel 8085 chip. As a minimum we were required to access information (reading data from the switch) and use this information to check for certain conditions. Based on the input information, appropriate output is to be displayed on the SDK-85. In general, the project is using I/O features to accomplish the task.

As the addition to the project, we connected 8-bit Digital-to-Analog Converter integrated chip that will convert digital data (input) into corresponding analog output. An Op-Amp is used to convert current analog output of the DAC to the voltage output that can be measured and compared. The program was developed and consists of three parts.

Part I (Input and Output)

This part of the program configures the ports 21 and 22 for the input and the output of data. Actual data transfer (Input and Output) is also done in Part I.

Part II (Voltage calculation)

From the data stored in the accumulator the converted voltage is calculated and stored in registers A and DE for display.

Part III (Display, Delay and CountDown)

In this part the calculated voltage is being displayed and delayed for 20 seconds. It then counts down from 10 to 0 seconds.

At this point program inputs new set of data and the process repeats itself again. While the program is in the delay and countdown process, position of switches can be changed to get different output voltage.

Essentially, the output voltage can be checked in three ways. First, by measuring the voltage potential across the output of the amplifier.

Second, the calculated voltage will be displayed on the SDK-85. Third, the output voltage can also be calculated using the equation:

$$V_{out} = \frac{(DigitalInput)_h}{256_d} \cdot V_{ref}$$

We used National Semiconductor DAC0830 integrated chip D/A converter that has current output. To convert analog current output of DAC to proportional voltage we used 741 Op-Amp.

Assembly Program

2000	LXI SP, 20C2	31		2027	INR H	24
2001		C2		2028	INR E	1C
2002		20		2029	INR E	1C
2003	MVI A, 02	3E		202A	PUSH PSW	F5
2004		O2		202B	MOV A, E	7B
2005	OUT 20	D3		202C	ADI, OO	C6
2006		2O		202D		OO
2007	IN 21	DB		202E	DAA	27
2008		21		202F	MOV E, A	5F
2009	OUT 22	D3		2030	POP PSW	F1
200A		22		2031	JMP "CMP H"	C3
200B	CALL **Voltage Conversion**	CD		2032		23
200C		1C		2033		20
200D		2O		2034	**** Voltage X. ****	OO
200E	CALL **Display**	CD		2035	LXI H, 2049	21
200F		6O		2036		49
2010		2O		2037		20
2011	CALL **Delay**	CD		2038	CMP M	BE
2012		7O		2039	JZ "MOV A, E"	CA
2013		2O		203A		44
2014	CALL **CountDown**	CD		203B		20
2015		8O		203C	JC "MOV A, E"	DA
2016		20		203D		44
2017	JMP 2007	C3		203E		20
2018		O7		203F	INR D	14
2019		20		2040	INX H	23
201A	**** VoltageConversion ****	OO		2041	JMP "CMP M"	C3
201B	**** Voltage .XX ****	OO		2042		38
201C	LXI D, 0000	11		2043		20
201D		OO		2044	MOV A, E	7B
201E		OO		2045	MOV E, D	5A
201F	MVI C, 00	OE		2046	MVI D, 00	16
2020		OO		2047		OO
2021	MVI H, 00	26		2048	RET	C9
2022		OO		2049	1 Volt reference	31
2023	CMP H	BC		204A	2 Volt reference	63
2024	JZ "Voltage X."	CA		204B	3 Volt reference	95
2025		35		204C	4 Volt reference	C7
2026		20		204D	5 Volt reference	F9

205E				2085	CALL UPDDT	CD
205F	**** Display ****			2086		6E
2060	PUSH PSW	F5		2087		O3
2061	CALL UPDAD	CD		2088	MVI C, O2	OE
2062		63		2089		O2
2063		O3		208A	LXI D, FE50	11
2064	POP PSW	F1		208B		50
2065	CALL UPDDT	CD		208C		FE
2066		6E		208D	CALL Delay	CD
2067		O3		208E		F1
2068	RET	C9		208F		O5
2069				2090	DCR C	OC
206A				2091	JNZ "LXI D"	C2
206B				2092		8A
206C				2093		20
206D				2094	POP PSW	F1
206E				2095	DCR A	3D
206F	**** Delay 20 seconds ****			2096	JNZ "PUSH PSW"	C2
2070	MVI C, 28	OE		2097		82
2071		28		2098		20
2072	LXI D, FE50	11		2099	RET	C9
2073		5O		209A		
2074		FE		209B		
2075	CALL Delay	CD		209C		
2076		F1		209D		
2077		O5		209E		
2078	DCR C	OD		209F		
2079	JNZ "LXI D"	C2		20A0		
207A		72		20A1		
207B		20		20A2		
207C	RET	C9		20A3		
207D				20A4		
207E				20A5		
207F	**** CountDown to OO****			20A6		
2080	MVI A, O9	3E		20A7		
2081		9		20A8		
2082	PUSH PSW	F5		20A9		
2083	MVI B, OO	O6		20AA		
2084		OO		20AB		

Data

Data IN (hex)	Calculated Voltage (V)	Measured Voltage (V)
00	$=00_h/256_d$ *5.10 V =0.00	0
FF	$=FF_h/256_d$ *5.10 V =5.10	5.10
7F	$=7F_h/256_d$ *5.10 V =2.55	2.58
80	$=80_h/256_d$ *5.10 V =2.55	2.54
F0	$=F0_h/256_d$ *5.10 V =4.80	4.84
E0	$=E0_h/256_d$ *5.10 V =2.24	2.84

Conclusion

This project encompassed all we have learned in the theory and from the previous experiments. For our program we used subroutines, delays, display of information on the Address/Data field of SDK-85 and various conditional and unconditional jumps.

After making all necessary connections, also shown on the Schematics page, and loading the program into memory we were able to run the program successfully. The program first accessed data (via port 21) and immediately displayed same binary sequence on the LED's (via port 22). The data also remained in the accumulator that was then used to calculate corresponding voltage. After the calculation part of the program, voltage was stored in the register A (.XX part of voltage) and register E (OX. part). Then using subroutines UPDAD and UPDDT we were able to display that voltage on the Address/Data field of the SDK-85. The process was first delayed for 20 seconds and then started to count down from 09 to 00 at which point the new set of data was acquired (IN instruction) and the program repeated. While the program is running and in the delay part, switch combination could be changed to get different voltage output. The output of the DAC is only changed after IN and OUT instructions were executed.

From the data we were able to obtain, we can say that we have accomplished the requirements of the I/O project. We were able to input the data, process it and then output the result in different forms.