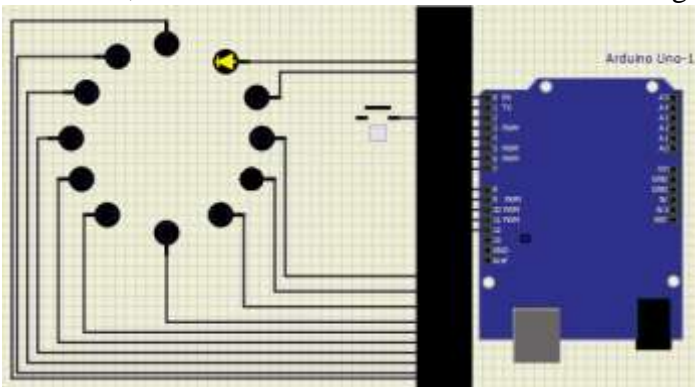
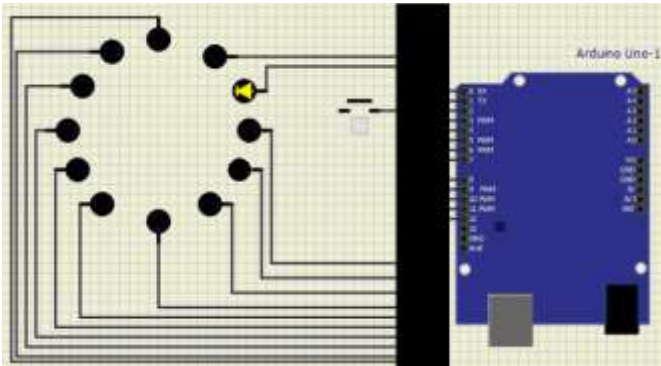




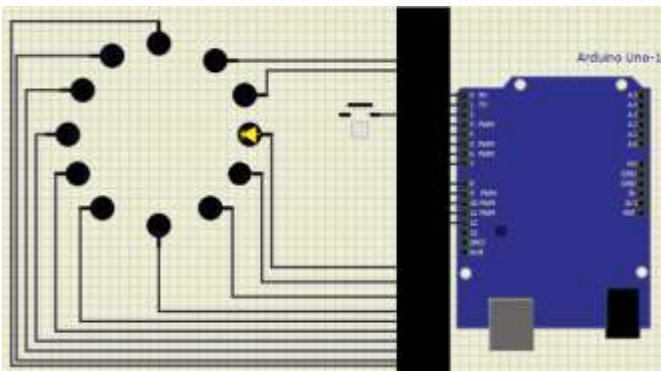
- 3- Therefore, after three seconds the next LED to the right is turned on while turning off the first LED.



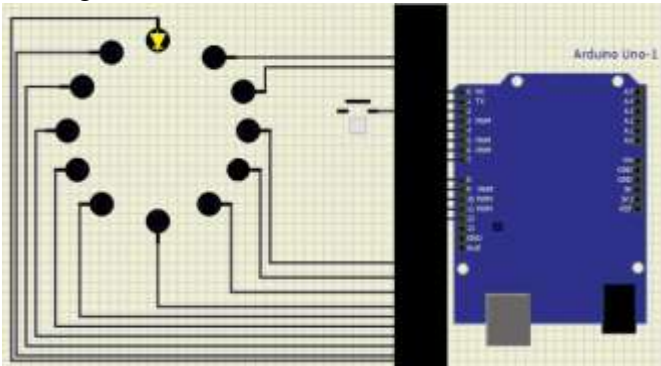
- 4- After another three seconds, the third LED is turned on.



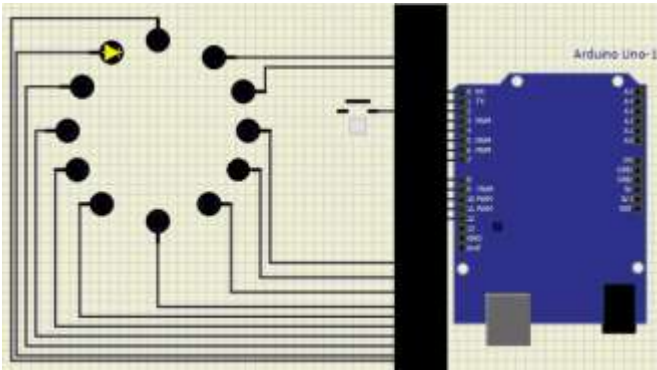
- 5- And so on...



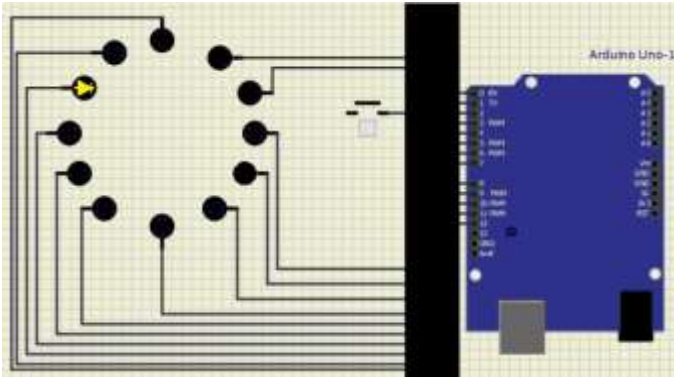
- 6- Pressing the button again, make the currently turned-on LED turn off immediately. The system starts turning on LEDs anticlockwise with 1 second between each one and the next.



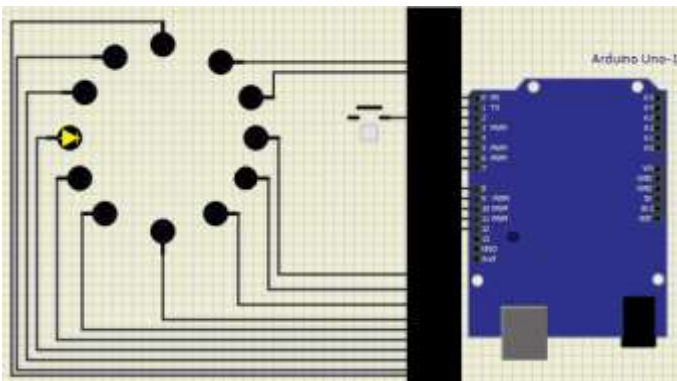
7- Therefore. After one second the first LED to the left is turned on while turning off the first LED.



8- After another one second, the second LED to the left is turned on while all other LEDs are turned off.



9- And so on...



10- Each time the button is pressed, the system changes LED sequence direction with three seconds between each LED and the next one in the clockwise case and one second for the anticlockwise case.

Notes:

1. The push button is pressed just for a **few milliseconds**.
2. Some connections are hidden or deleted in the previous screens.
3. You are not allowed to use any external library. All “#include lines” in your code will be deleted automatically before task evaluation.
4. Use the attached “task2.simu” file as a start for your design. You can add components, change components properties but don’t change components arrangement. (arrangement change = -2 Marks)

Grading criteria:

1 Mark: The circuit is connected correctly and the system starts with all LEDs switched off.

If it doesn’t **always** respond to the push button as it pressed then no more marks. Otherwise,

1 Mark: turning it clockwise

1 Mark: with 3 seconds between each one and the next

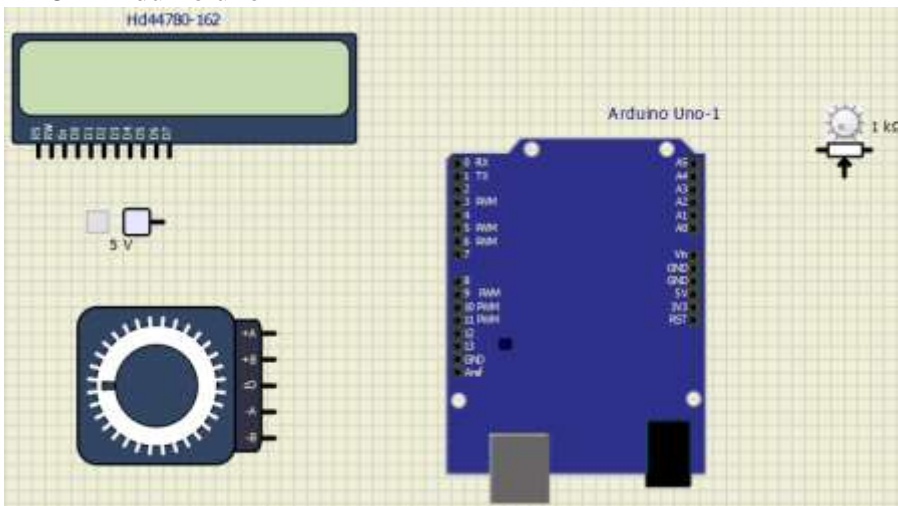
1 Mark: turning it anticlockwise

1 Mark: with 1 second between each one and the next

Task 3: kitchen hood (5 Marks)

It is required to build an automatic kitchen ventilation fan (مروحة تهوية) that works also as kitchen hood (شفاف مطبخ). If there is smoke it works as a kitchen hood to absorb it. If there is no smoke and the temperature is more than 25 degrees, it works as a fan. Otherwise, the motor is stopped. An always on LCD is used to display the current temperature. This system contains

- 1- A stepper motor that is rotating clockwise with 2 rpm to work as a fan and anticlockwise with 4 rpm to absorb smoke. For simplicity, no need for a motor driver.
- 2- Digital smoke detector sensor to detect smoke. Zero means no smoke while one indicates smoke existence. It is replaced by a tact switch (“Fixed Volt” in the SimuLIDE) for simulation. If there is smoke, the motor rotates anticlockwise to start absorbing smoke. The motor keeps rotating anticlockwise for extra 3 seconds after there is no smoke (the tact switch is tact low)
- 3- LM35 temperature sensor. To sense the kitchen temperature. It is replaced by potentiometer.
- 4- LCD to display the current temperature.
- 5- Arduino uno



You are not allowed to use any #include except LiquidCrystal.h (stepper.h is not allowed. You can rotate it yourself – Lecture 3). You can check the “LCD_example.ino” file included with this task for how to use the LCD.

Grading criteria:

1 Mark: The circuit is connected correctly.

1 Mark: LCD *always* displays the current temperature correctly with the potentiometer change.

If the motor doesn't respond to the tact switch (smoke sensor) – even if it is high for a few milliseconds then being low again – then no more marks. Otherwise,

1 Mark: the motor is rotating clockwise with 2 RPM

1 Mark: the motor is rotating anticlockwise with 4 RPM

1 Mark: the motor keeps rotating for an extra 3 seconds after the tact switch is tact low.

Task 4: Programmable Arduino (5 Marks)

Instead of programming Arduino uno to do certain tasks, it is required to program Arduino uno to be user programmable. You need to develop a very simple programming language. A 4 X 4 keypad is connected to 8 pins of the Arduino containing numbers from 0 to F. To start programming the user has to press 0. You allow the user to use two internal variables Var1 and Var2. Two digital output pins connected to two LEDs Yellow and Red, All pins

Used devices

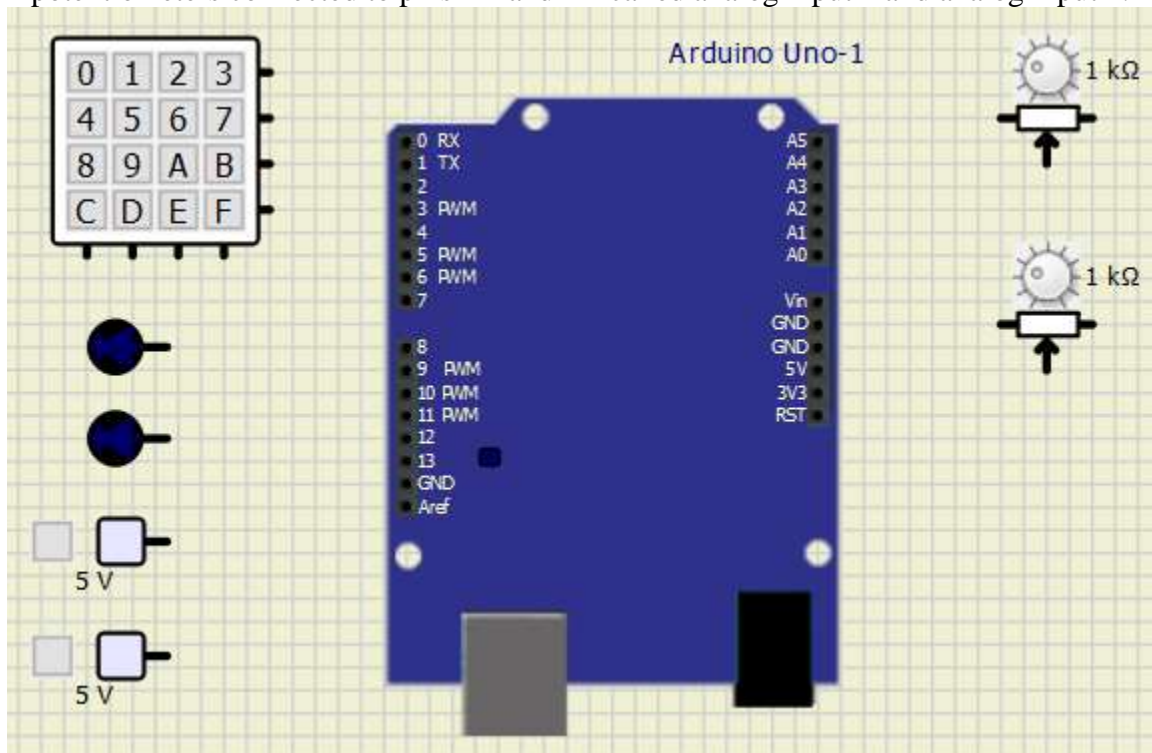
Arduino uno

4x4 keypad

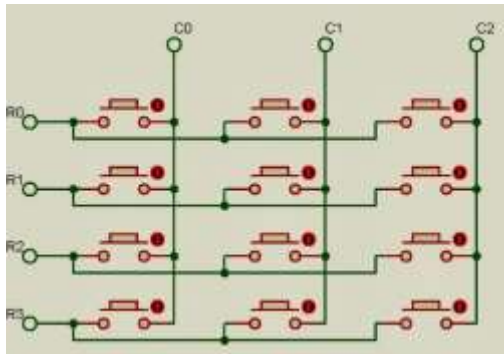
2 LEDs (Yellow and Red) the yellow LED is connected to pin 8 and the Red LED is connected to pin 9. Called output 1 and output 2.

2 tact switches (“Fixed Volt” in the SimuLIDE) – Switch 1 is connected to pin 10 and switch 2 is connected to pin 11. Called input 1 and input 2.

2 potentiometers connected to pins A1 and A2 called analog input 1 and analog input 2.



Note: This is the internal circuit of the keypad



This programming language rules are as the following:

Op code	Command	Description	Example	input sequence (from left to right)
0	Start Coding	At the beginning of the program. Any keys pressed before pressing 0 are discarded except "D". Note: pressing 0 during the program coding doesn't restart writing the program. It is just used at the beginning of the program (Check "D")	Start Coding	0
1	ADD Variable, Value	ADD a value (between 0 and 9) to a variable	add var1,6	116
			add var2,4	124
2	MOV Variable, Value	Put a value (between 0 and 9) in a variable	mov var1,7	217
			mov var2,3	223
3	ADD Variable, Variable	Add a variable to another variable	add var1, var2	312
			add var2, var1	321
4	MOV Variable, Variable	Put the value of a variable in the other variable	mov var1, var2	412
			mov var2, var1	421
5	If (var1 [operator] value)	Apply if condition on variable 1 to compare it with a value between 0 and 9. three possible comparison operators are supported. Greater than (>) coded as 1, Less than (<) coded as 2 and equal (==) coded as 3. based on the comparison, only one command after the if statement is executed or not.	if var1>5	515
			if var1<6	526
			if var1==7	537
			if var1>5 mov var1,3	515213
			if var1<6 add var1,2	526112
			if var1==7 add var1, var2	537312
6	If (var2 [operator] value)	Just like opcode 5 but for variable 2 instead of variable 1	if var2>5	615
			if var2<6	626
			if var2==7	637
			if var2>5 mov var1,3	615213
			if var2<6 add var1,2	626112
			if var2==7 add var1, var2	637312

Op code	Command	Description	Example	input sequence (from left to right)
7	If (variable [operator] variable)	Just like opcodes 5 and 6 but for the comparison between two variables. It is encoded as 7 followed by the operator code followed by the first variable in the comparison (the leftmost variable)	if var1>var2	711
			if var1<var2	721
			if var1==var2	731
			if var2>var1	712
			if var2<var1	722
			if var2==var1	732
8	Loop N times	Repeat the following instruction N times. For example, Loop 5 times will repeat the following instruction 5 times. It repeats just one instruction and N is between 0 and 9. it is coded as 8 followed by zero (always zero just to make this instruction 3 digits) followed by N.	Loop 5	805
			Loop 7	807
			Loop 4	804
9	Digital write one/zero to output pin 8 or 9	The yellow LED is connected to pin 8 let's name it output 1 and the Red LED is connected to pin 9. let's name it output2. This command is used to turn these LEDs on or off. It is coded as 9 followed by the value zero or one followed by the output number one or two.	write zero (turn off) to output1 (pin 8)	901
			write one (turn on) to output1 (pin 8)	911
			write zero (turn off) to output2 (pin 9)	902
			write one (turn on) to output2 (pin 9)	912
A	Digital read var1 or var2 from input pin 10 or 11	The two tact switches are connected to pins 10 let's name it input 1 and pin 11. let's name it input 2. This command is used to read from these tact switches and put the input value in one of the two variables. It is coded as "A" followed by the variable number (one or two) followed by the input number one or two.	read variable 1 from input1 (pin 10)	A11
			read variable 2 from input1 (pin 10)	A21
			read variable 1 from input2 (pin 11)	A12
			read variable 2 from input2 (pin 11)	A22
B	Analog read var1 or var2 from analog input pins A1 or A2	The two potentiometers are connected to pins A1 let's name it analog input 1 and pin A2 let's name it analog input 2. This command is used to read from these potentiometers and put the input value in one of the two variables. It is coded as "B" followed by the variable number (one or two) followed by the input number one or two.	read variable 1 from analog input1 (pin A1)	B11
			read variable 2 from analog input1 (pin A1)	B21
			read variable 1 from analog input2 (pin A2)	B12
			read variable 2 from analog input2 (pin A2)	B22

Op code	Command	Description	Example	input sequence (from left to right)
C	Cancel the last pressed key.	if the user pressed any key by mistake, s/he can press "C" after this key to cancel the last entered key. Pressing "C" twice, clears the last entered two keys and so on...	add var1,6 (if pressed "2" key by mistake)	112C6
D	Clear all entered keys and start writing the program again.	Clear all entered keys to restart writing the code from the beginning. No need for pressing "zero" again after "D" to start writing the program. The user is able to start writing the program immediately after pressing "D". Pressing "D" at the beginning of writing the code is equivalent to pressing zero.	add var1,6 (after many pressed keys)	1250512522A11D116
E	Has no meaning.	Has no meaning. Just discard it. it is just like pressing F5 or refreshing on your desktop 😊.	add var1,6 (after many pressed keys)	1250512522A11D1EE1E6
F	Execute the program	After the user completes writing his/her program. s/he must press "F" to start the program execution.	Check the complete programs examples	

Examples:

- 1- If the user entered 0911F then output1 (pin 8 - yellow LED) should be turned on.
- 2- If the user entered 125420A115309E11531900C1F the system function will be as the following:

Input Code	Meaning	Notes
12542	discarded because anything before the first zero is discarded.	
0	Start Coding	This is the first meaningful input
A11	read variable 1 from input1 (pin 10)	
530	if var1==0 (LOW)	
9E11=911	write one (turn on) to output1 (pin 8)	E is just discarded
531	if var1==1 (HIGH)	
900C1=901	write zero (turn off) to output1 (pin 8)	C removes the last input digit. Therefore, the second zero is removed.
F	Execute the program	

Meaning the in this case the system will act exactly like the “simple LED” system described on task 1

- 3- If the user entered
5426A50217223116911D210B21803312519911529912519902529901539901539902F
the system function will be as the following:

Meaning	Input Code	Notes
5426A5	discarded	because anything before the first zero is discarded
0	Start Coding	This is the first meaningful input
217	mov var1,7	
223	mov var2,3	
116	add var1,6	
911	write one (turn on) to output1 (pin 8)	This LED shouldn't turn on at this line because this command will be canceled in the next command.
D	Clear all entered keys and start writing the program again.	discard everything before this line. This is the actual entry point of this program.
210	mov var1,0	put zero in variable 1
B21	read variable 2 from analog input1 (pin A1)	and read the second variable from the analog input (it ranges between 0 and 1023)
803	Loop 3	add the read input to variable 1 three times.
312	add var1, var2	
519	if var1>9	
911	write one (turn on) to output1 (pin 8)	if variable 1 > 9 turn on the yellow LED
529	if var1<9	if variable 1 < 9 turn on the Red LED. Note: both LEDs are turned off it is equal to 9
912	write one (turn on) to output2 (pin 9)	
519	if var1>9	If you are turning on pin 8 then turn off pin 9
902	write zero (turn off) to output2 (pin 9)	
529	if var1<9	If you are turning on pin 9 then turn off pin 8
901	write zero (turn off) to output1 (pin 8)	
539	if var1==9	If var 1 == 9 turn off both LEDs
901	write zero (turn off) to output1 (pin 8)	
539	if var1==9	
902	write zero (turn off) to output2 (pin 9)	
F	Execute the program	Start executing the program

In this case, after pressing the execution button (F), the system will keep reading the potentiometer value. If it is more than 3 then only the yellow LED will be turned on. If the potentiometer value is less than 3 then only the red LED will be turned on. If it is equal to 3 then both of them will be turned off.

You are not allowed to use any #include (Keypad.h is not allowed). You can implement a key detection / identification algorithm yourself. (Lecture 1)

Grading criteria:









1 Mark: for each working example = 1Mark * 3 = 3 Marks

2 Mark: two more hidden examples will be applied = 1Mark *2 = 2 Marks

Tasks' submission:

- You must submit your tasks as 16 files. Four files for each task (.ino - .hex - .simu - .mp4 [or any other valid video format]) named YourCode_taskNumber_VersionNumber.extension. If you are a semester student your code is 7 or 8 digits starting with "9" (You login to the faculty site using this code). If you are a credit hours student, your code is 7 or 8 digits starting with 1,2,3 or 4. For example, if your code is "1212121" your submitted files must be like
 - 1212121_1_V3.ino
 - 1212121_1_V3.simu
 - 1212121_1_V3.hex
 - 1212121_1_V3.mp4
 - 1212121_2_V5.ino**
 - 1212121_2_V5.simu**
 - 1212121_2_V5.hex**
 - 1212121_2_V5.mp4**
 - 1212121_3_V1.ino
 - 1212121_3_V1.simu
 - 1212121_3_V1.hex
 - 1212121_3_V1.mp4
 - 1212121_4_V8.ino**
 - 1212121_4_V8.simu**
 - 1212121_4_V8.hex**
 - 1212121_4_V8.mp4**

Like this

<input type="checkbox"/> Name	Date modified	Type
 102030_1_V27.ino	5/16/2021 8:48 PM	Arduino file
 102030_1_V27.simu	5/16/2021 8:48 PM	SIMU File
 102030_2_V1.ino	5/16/2021 8:48 PM	Arduino file
 102030_2_V1.simu	5/16/2021 8:48 PM	SIMU File
 102030_3_V7.ino	5/16/2021 8:48 PM	Arduino file
 102030_3_V7.simu	5/16/2021 8:48 PM	SIMU File
 102030_4_V9.ino	5/16/2021 8:48 PM	Arduino file
 102030_4_V9.simu	5/16/2021 8:48 PM	SIMU File

- The .mp4 file for each task is no more than 3 minutes video with your voice describing your code idea and showing all testcases.
- Any other naming format like "task1.ino", "my task.ino", "1212121_ahmed_ali task1.ino" ... etc will be deleted automatically and will not be considered as a submission.
- If you want to add any comments about your solution, add it as a comment at the begin of the .ino file
- Don't write your name or your code inside any of the files, your code should appear only at the file name.
- If you need to re-submit your work, change V1 in the file name to be V2, V3 ... etc. For example, if you are resubmitting for the first task for the third time, your file name must be YourCode_1_V3.ino and YourCode_1_V3.simu
- Submit your files to the following form. You must upload all files. If you will not submit one of them, upload an empty file instead adding extension ".empty" for example "102030_1_v1.ino.empty" to continue submission.
<https://forms.gle/4XMdX8F9AG8MAbnV6>

Penalties:

-20 for cheating [Copied designs or codes/ code parts from someone else or from the internet]. Therefore, please don't use the internet or talk with anyone else about the tasks.

-1 / 3 minutes: for the delayed submission after the deadline. Submission will be closed