T.C.
DOKUZ EYLUL UNIVERSTY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

2022 – 2023
SPRING SEMESTER

CME 3208
PRINCIPLES OF EMBEDDED SYSTEMS

LAB 7:
PARKING SENSOR

DUE DATE:
23:55 – 25.04.2023

In this lab, you are asked to create a parking sensor, using HC-SR04 Ultrasonic Sensor, LCD display, buzzer and a LED. You will use an ultrasonic sensor to calculate distance, print it out on LCD display and activate the buzzer and LED according to ultrasonic sensor value you read.

The calculation of buzzer and LED activation frequency should be done according to variables and equation below.

These variables are for activation duration for buzzer and LED. We assume that buzzer and LED should activate up to 1000 milliseconds and deactivate for same time. However, according to our formula below, calculated new frequency could exceed "BUZZER_LED_FREQ_MAX" variable.

```
int BUZZER_LED_FREQ_MAX = 1000;
int BUZZER_LED_FREQ_MIN = 10;
```

These variables control the distance space we use on our equation. You can consider them as centimeters, to make it easier to program. However, our function could work if the distance is larger than 1 meter, giving a larger activation frequency than "BUZZER_LED_FREQ_MAX" variable.

```
int DIST_MAX = 100;
int DIST_MIN = 10;
```

These variables are used to store current distance returned by sensor and the result of current frequency calculation.

```
int CURRENT_DIST = 0;
int CURRENT_FREQ = 0;
```

These variable are used to how frequently we should check the current distance and update "CURRENT_DIST" variable and LCD display for values. You can assign these values but we should be able to check the behavior for different update values on your project.

```
int UPDATE_FREQ_BUZZER = 200;
int UPDATE_FREQ_LCD    = 200;
```

The formula below are used to calculate what the current frequency should be according to current distance measured by sensor.

$$Current_{Freq} = \left(\left(\frac{Current_{Dist} - Dist_{Min}}{Dist_{Max} - Dist_{Min}}\right) * \left(BuzzerLEDFreq_{Max} - BuzzerLEDFreq_{Min}\right)\right) + BuzzerLEDFreq_{Min}$$

As you can see, every “UPDATE_FREQ_BUZZER”, “CURRENT_FREQ” is calculated and buzzer and LED are activated according to this variable. For out of bound cases where distance is smaller than 10 cm and larger than 100 cm, we can try to implement special rules. We can say that for distance values lower than “DIST_MIN”, the buzzer and LED should be activated without time limit, until “CURRENT_DIST”is larger than “DIST_MIN”. We can also say that for distances larger than “DIST_MAX”, the program should not activate buzzer or LED, because it is not necessary.

If you wish you can change the formula or the values of variables above to make your program work better. That will allow us to see which formula and which initial values are better for this project.

You should make sure that you do not get a zero division or other mathematical errors on your formula. Include checks to prevent such scenarios.

You should print “CURRENT_DIST” and “CURRENT_FREQ” on your LCD and it should be updated according to “UPDATE_FREQ_LCD” variable. An example LCD display is given below.

DEFAULT LCD SCREEN:

	00	**01**	**02**	**03**	**04**	**05**	**06**	**07**	**08**	**09**	**10**	**11**	**12**	**13**	**14**	**15**
00	D	I	S	T	A	N	C	E		:			5	0	c	m
01	F	R	E	Q	U	E	N	C	Y	:		4	5	0	m	s

For measured distance 50 cm, “CURRENT_FREQ” is calculated to be 450 milliseconds according to the formula above. This means the buzzer and LED will activate for 450 milliseconds and deactivate for 450 milliseconds.

But what happens if we update the “CURRENT_DIST” variable due to “UPDATE_FREQ_BUZZER” variable? Should we wait until this buzzer and LED cycle ends or should we reset it according to “UPDATE_FREQ_BUZZER ”? If we wait for it to end, our program will not be responsive as we wish it to be. If we do not wait, it may cause buzzer activation without silent period due to frequent changes in “CURRENT_DIST”. Try to implement the second case and if you can find a solution to make it more user friendly, like a real life park sensor, explain it in our lab class.

Make sure to use your buzzers in a quieter fashion, rather than very loud whistling sound. You can use potentiometer to regulate sound level of buzzers.

If you have problems with LCD display, try to use serial monitor as an alternative. It will allow you to show your results but your grade would still be reduced due to not using a LCD display for your lab assignment.

UPLOAD REQUIREMENTS:

You are required to upload three different files for this assignment. First is a text file that contains your source code for your circuit. Second is a Tinkercad screenshot showing your circuit after your design is finished. Third is a video file showing your circuit that has been created using lab supplies. Only one of the group members can make an upload, there is no requirement for all members to do a separate upload of same files.

For your video, you are expected to introduce yourself and your teammates, showing your student IDs in video too. If your group are working online rather than meeting in real life, you do not have to show the ID cards of students who are not physically present.

You are also expected to explain your circuit in a brief but complete manner and show its execution. This video should not be too long (a maximum of 3 to 5 minutes is acceptable) and should not be too large otherwise you will not be able to upload to Sakai (check the maximum allowed file size for upload to Sakai). You should also consider choosing resolution 720p over 1080p or larger for smaller file size (make sure your circuit is still visible and understandable over this resolution). You can also change your video format to a more space efficient one.

You should show the effects of changing every global variable that is mentioned above in your video. Again, please do not consider too many different values to make a too detailed video, just show the effect of changing that particular global variable.

If you experience problems with your circuit and cannot make it work, show your design in Tinkercad and your circuit at the same and explain what is the problem and what might be causing it in your opinion.

The naming of your files should follow the format below for 3 person groups. For 2 or 4 person groups just write 2 or 4 student numbers in ascending order. You should write your group numbers with one leading zero if it is between 1 and 9, normally for 10 and above.

The file extension for screenshot (Tinkercad) and video does not have to be “jpg” and “mp4”, you can use other formats as well like “png” or “mkv” or etc. However, make sure you use commonly used image and video formats, if we cannot open it on our computers, you grade related to them will be zero. In addition your source code file extension could be either “c” or “ino”, do not leave it as “txt”.

Do not “zip” or “rar” requested files and upload them. It is not necessary and it makes it harder for us to evaluate your assignments. Please upload 3 files as they are without compressing them to a single file.

FORMAT:
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>_CODE.c
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>_TINKERCAD.jpg
GROUP_<group_number>_<student_number_1>_<student_number_2>_<student_number_3>_VIDEO.mp4

EXAMPLE:
GROUP_01_2023510123_2023510124_2023510125_CODE.c
GROUP_01_2023510123_2023510124_2023510125_TINKERCAD.jpg
GROUP_01_2023510123_2023510124_2023510125_VIDEO.mp4

You are expected to write your own code for algorithms instead of using an available method for calculations. If you use such as a method that makes this assignment trivial to code, your grade for coding will be zero.

Your uploaded source codes (as well as your circuit design) will be checked for cheating and plagiarism. If cheating is detected, your entire assignment will be graded zero. If you or other students copy your code from an online source rather than writing it yourself, it will be considered as cheating as well.

Make sure that you upload your correct assignment to correct upload section. If you accidentally upload another assignment (from another class for example) or to an incorrect upload (other section's upload), it will be considered as not turned in and it will be graded as zero. Worst of all, you will only realize it after grades are published and it will too late to fix it.

If you have any questions or problems regarding this lab paper, you can ask about it in our lab sessions. If you wish, you can also ask it in class forums or assignment page comments. If you send an email and if your question is answered, please share this information with other students to prevent asking of the same question again and again.

Your assignment will be open for upload until 23:55, 26.04.2023. This is done to allow students who may experience extreme problems (no Internet or electricity, computer crash or failure, etc.) and miss the deadline as a result. This one day extension will allow them to upload. If you are still unable to upload, send us an email informing your situation and at the same time, try everything you can to make your assignment upload.

Lastly, please do not forget to click "Submit" button after you upload your assignment files. If you do not, even though your files are uploaded to Sakai, you are labeled as "No Submission" and ignored when we try to download your assignments, making your uploaded files invisible to us, leading us to assume you did not make an assignment submission.

GOOD LUCK TO YOU ALL!