

CS 431-1 Embedded Systems

[Dashboard](#) / [My courses](#) / [CS 431-1](#) / [25 October - 31 October](#) / [HW 3](#)

HW 3

Opened: Wednesday, 29 September 2021, 12:00 AM

Due: Friday, 12 November 2021, 5:59 PM

To do: Make a submission

Working Draft! - Keep checking here for changes.

Your assignment is to build a morse code flasher. You shall program the edsim51 emulator to receive characters from serial port and "flash" the corresponding chars using the DAC.

"Flashing" via the DAC means: sending a nonzero number -of your choice- representing short-pulses and sending another number -of your choice- representing long-pulses. You should also keep this DAC value for a predefined length of time and that it should be brought down to zero when no long or short pulses are sent. This length of time would be somewhere around 0.5-1 seconds in reality but find a more practical number yourselves for demo purposes (i.e. *simulator time is about 1/100th of live-time, so maybe flash for 100ms instead of 1s*). Also note that there should be a zero volt period between pulses (*short-to-short, short-to-long, long-to-short, long-to-long*). All this shall be visible in the DAC graph (the DAC plot X-axis is fixed so it should also give you some idea about what length-of-time to choose)

- Input chars should be buffered internally until enter (\n) is sent from serial (fixed size buffer is OK). **This is why you must use edsim51_nonewline variant!**
- Chars that are not numbers should be ignored (e.g. abc123qwe is equivalent to 123). So you only need to write a morse lookup table for chars '0'-'9'.
- If enter was pressed during the flashing of a previous sequence, the flashing job should restart with the new sequence.
- Make sure to think of and to handle any other potential problems.

You will make a demo of, and upload a code of working parts below that build up to the full functionality. No report is needed, however you will demo your code and present it to myself or your TA.

To begin with, you will use Round-Robin architecture (no ISRs (callbacks, etc.)).

Parts:

1. Demonstrate you are able to receive chars by echoing them back. (5 pts)
2. Demonstrate you are able to save sequences by printing them back when enter is pressed. (10 pts)
3. Demonstrate you are able to flash certain hard-coded chars. (10 pts)
4. Demonstrate you are able to flash hard-coded strings. (5 pts)
5. Demonstrate you can report back to serial when you finish flashing strings. (5 pts)
6. Demonstrate you can report back to serial when you cancel flashing strings; i.e. when a new sequence is entered before flashing a new one is finished. (5 pts)
7. Connect it all together, and demonstrate full functionality. i.e. demonstrate you can cancel flashing of sequences and handle other test cases. (10 pts). Cancellation does not need to occur during a character's flashing.
8. Repeat all with RR w/ISR architecture (40 pts).

In-demo questions will be worth the last 10 pts.

For this lab only; for morse-code sequences programming busy-waits are allowed. But please feel free to use Timers or any other method of non-busy waiting if you feel like it, you already have the code segments ready from previous lab. Also note that you are only allowed to use official keil libraries and not the user/community libs/codes.

p.s.1: some of you may notice that some of these steps are basically the same code and/or 1 line of difference. That is fine, it just means you were on the right path and it just means free points for you.

p.s.2: I may re-cycle this lab assignment and ask you to re-implement almost entirely the same system using an RTOS (FreeRTOS) as the final lab. Do this lab well, and you'll be able to recycle a lot of code.

For this lab it's utmost important that you use the edsim51_nonewline.jar variant, the standard one doesn't play nice with getc, getchar or scanf variants. The standard one sends a new line every time you press send.

Submission status

Submission status	No attempt
Grading status	Not graded
Time remaining	1 day 21 hours
Last modified	-
Submission comments	▶ Comments (0)

[Add submission](#)

You have not made a submission yet.

[◀ \(Cyber\)security of Embedded Systems](#)[HW 3 Discussion ▶](#)You are logged in as Pinar Yücel ([Log out](#))[Reset user tour on this page](#)

CS 431-1

[My Dashboard](#)[Bilkent Moodle Services](#)[Previous semesters \(Centrum\)](#)[General Purpose \(gen3Moodle\)](#)[Get Support](#)[Moodle@Bilkent Tutorials](#)[moodle.org Resources](#)[Zoom@Bilkent Tutorials](#)[zoom.us Resources](#)[BETS Guidelines](#)[BETS Workshops, Seminars](#)[Units/Services](#)[Educational Technology Support Unit \(BETS\)](#)[Bilkent Computer Center \(BCC\)](#)[STARS](#)[SRS](#)[AIRS](#)[Webmail](#)[Bilkent Home](#)[Academic Calendar](#)[Bilkent News](#)[English \(en\)](#)[Deutsch \(de\)](#)[English \(en\)](#)[Español - Internacional \(es\)](#)[Français \(fr\)](#)[Italiano \(it\)](#)

[Türkçe \(tr\)](#)[Русский \(ru\)](#)[العربية \(ar\)](#)[한국어 \(ko\)](#)[日本語 \(ja\)](#)[简体中文 \(zh_cn\)](#)[Data retention summary](#)[Get the mobile app](#)**moodle**[Previous Semesters](#)[General Purpose \(gen3Moodle\)](#)[Moodle@Bilkent Tutorials](#)[Zoom@Bilkent Tutorials](#)[Guidelines by BETS](#)[Ed. Tech. Support Unit \(BETS\)](#)[Bilkent Computer Center \(BCC\)](#)[STARS --- SRS --- AIRS](#)[Webmail](#)[Bilkent Home](#)[Academic Calendar](#)[Bilkent News](#)