



Embedded pool

Module09 : 7 segments display

contact@42chips.fr

Summary: When GPIOs are not enough

Version: 1

Chapter I

Préambule

Expanders are used in electronics to extend the capabilities of a microcontroller in terms of the number of inputs/outputs (I/O).

They allow multiple peripherals to be connected to a single I2C bus without taking up too many pins on the microcontroller.

This is useful for projects that require many inputs/outputs but do not have enough pins available.

I2C expanders are also very convenient because they can be easily configured and controlled using simple I2C commands, making project development faster and easier.

Chapter II


General instructions

Unless explicitly stated otherwise, the following instructions will be valid for all assignments.

- The language used for this project is C.
- It is not necessary to code according to the 42 norm.
- The exercises are ordered very precisely from the simplest to the most complex. Under no circumstances will we consider or evaluate a complex exercise if a simpler one is not perfectly successful.
- You must not leave any files other than those explicitly specified by the exercise instructions in your directory during peer evaluation.
- All technical answers to your questions can be found in the **datasheets** or on the Internet. It is up to you to use and abuse these resources to understand how to complete your exercise.
- You must use the datasheet of the microcontroller provided to you and comment on the important parts of your program by indicating where you found the clues in the document, and if necessary, explaining your approach. Don't write long blocks of text, keep it clear.
- Do you have a question? Ask your neighbor to the right or left. You can ask in the dedicated channel on the Piscine's Discord, or as a last resort, ask a staff member.

Chapter III

Finger display

	Exercise 00
A glimmer of hope (again...)	
Turn-in directory : <i>ex00/</i>	
Files to turn in : <i>Makefile, *.c, *.h</i>	
Allowed functions : <i>avr/io.h, util/delay.h</i>	


I am going to ask you to simply turn on a LED for this exercise. And since it's D(ay)9, you'll have to make the D9 LED blink at 1Hz. Unfortunately, the D9 LED is not directly connected to the MCU. Now that you're all grown up, it's up to you to find a solution!



A little look back at the schema looks necessary !


Chapter IV

Easy

	Exercise 01
Binary counter (again ?)	
Turn-in directory : <i>ex01/</i>	
Files to turn in : <i>Makefile, *.c, *.h</i>	
Allowed functions : <i>avr/io.h, util/delay.h</i>	

Great! You managed to use U4 in output mode. Now let's also test the inputs.
You need to write a program that:

- Every time you press the SW3 button, increment a counter.
- And constantly display its value on the D9 D10 D11 LEDs in binary.


	Exercise 02
Display a number	
Turn-in directory : <i>ex02/</i>	
Files to turn in : <i>Makefile, *.c, *.h</i>	
Allowed functions : <i>avr/io.h, util/delay.h</i>	

Now let's focus on the 7-segment display.

- You need to write a program that displays "2" on the rightmost digit.
- The other 3 digits should not be lit.


Chapter V

Medium

	Exercise 03
Papa Beaver's Storytime	
Turn-in directory : <i>ex03/</i>	
Files to turn in : <i>Makefile, *.c, *.h</i>	
Allowed functions : <i>avr/io.h, util/delay.h</i>	

You need to write a program that displays a number on the rightmost digit.

- This number should increment every second.
- When the number exceeds 9, it should return to 0.
- The other 3 digits should not be lit.

	Exercise 04
Double agent	
Turn-in directory : <i>ex04/</i>	
Files to turn in : <i>Makefile, *.c, *.h</i>	
Allowed functions : <i>avr/io.h, util/delay.h</i>	

Now you have to display "42" on the rightmost digits.


- It may seem simple, but it's not!
- The other 2 digits should not be lit.




The two numbers must be visible at the same time to a human being without any "visible" [flickering](#)

Chapter VI

Very hard (:

	Exercise 05
	The magical bus
	Turn-in directory : <i>ex05/</i>
	Files to turn in : <i>Makefile, *.c, *.h</i>
	Allowed functions : <i>avr/io.h, util/delay.h</i>

- You need to write a program that displays a number on all 4 digits.
- This number should increment every second.
- When the number exceeds 9999, it should return to 0.

	Exercise 06
	Twitter
	Turn-in directory : <i>ex06/</i>
	Files to turn in : <i>Makefile, *.c, *.h</i>
	Allowed functions : <i>avr/io.h, util/delay.h</i>

- You need to retrieve the value of RV1 on 10 bits.
- And display it on the 7-segment display.



When the ADC returns 0, you must display "0000"

When the ADC returns 42, you must display "0042"

When the ADC is at its maximum, you must display "1023"