POLITECNICO
DI MILANO

www.polimi.it

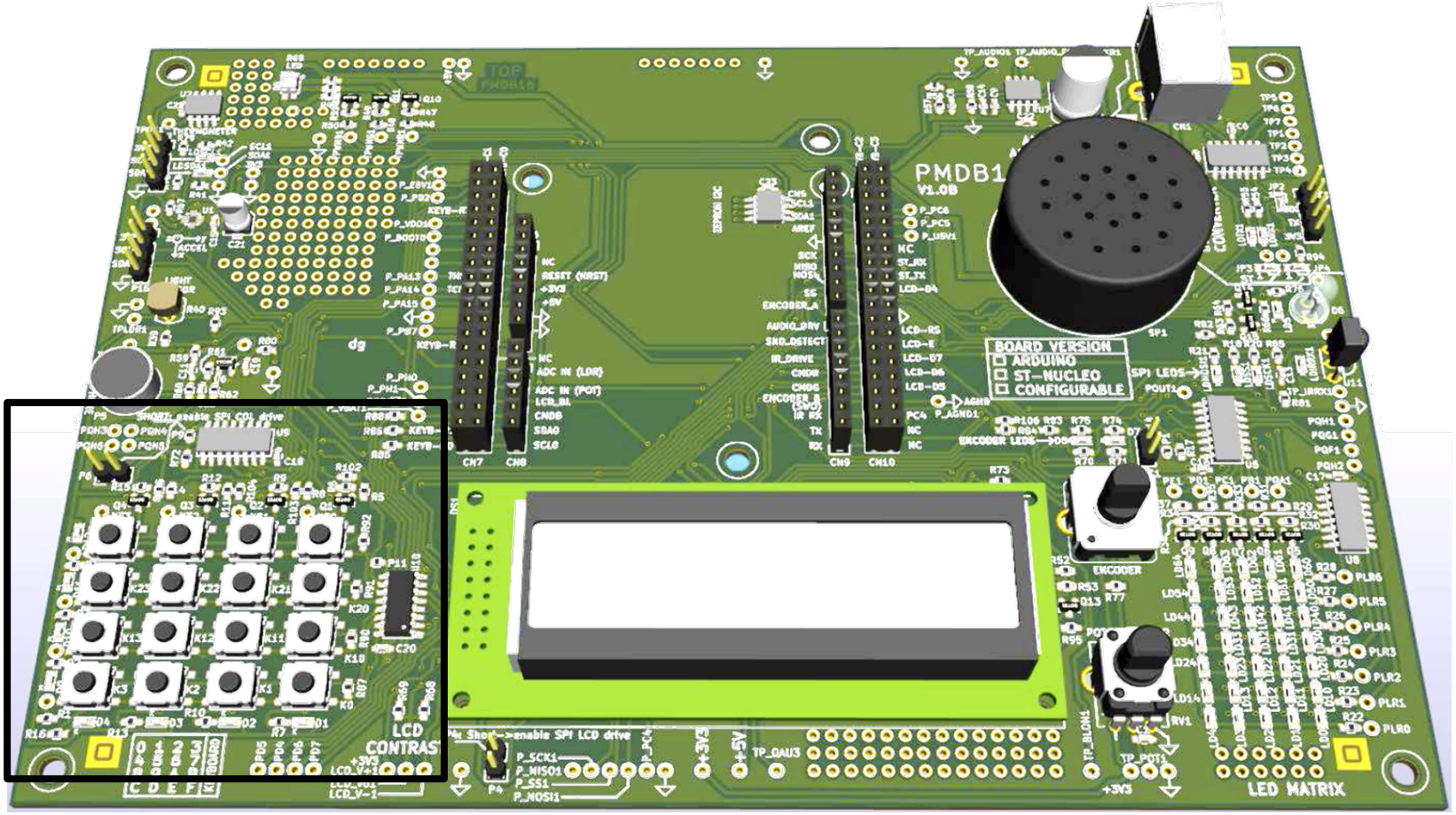




STM32 – KEYBOARD

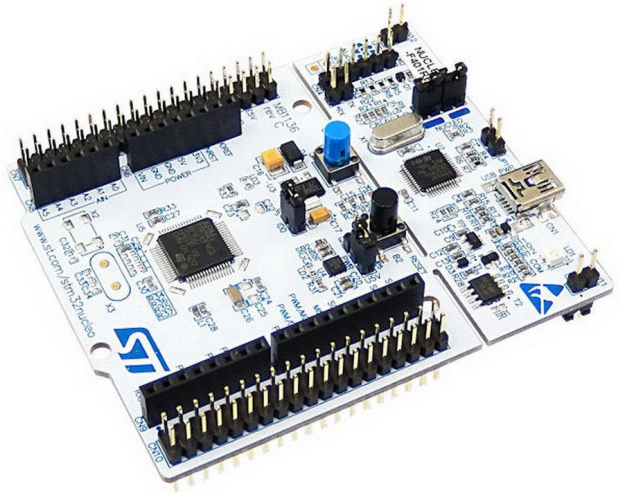
Dr. Federica Villa

KEYBOARD– *Evaluation Board Components Overview*

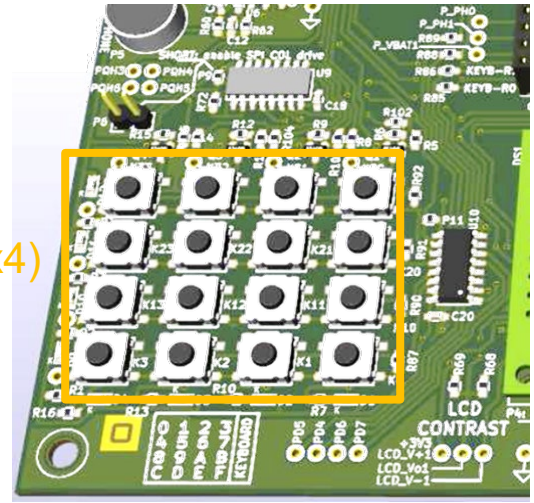


KEYBOARD

KEYBOARD – *PMDB16 details*



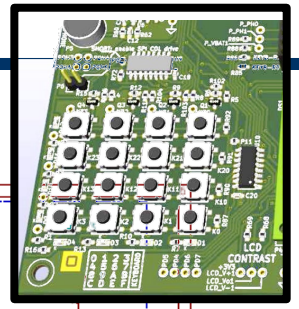
KEYBOARD (4x4)

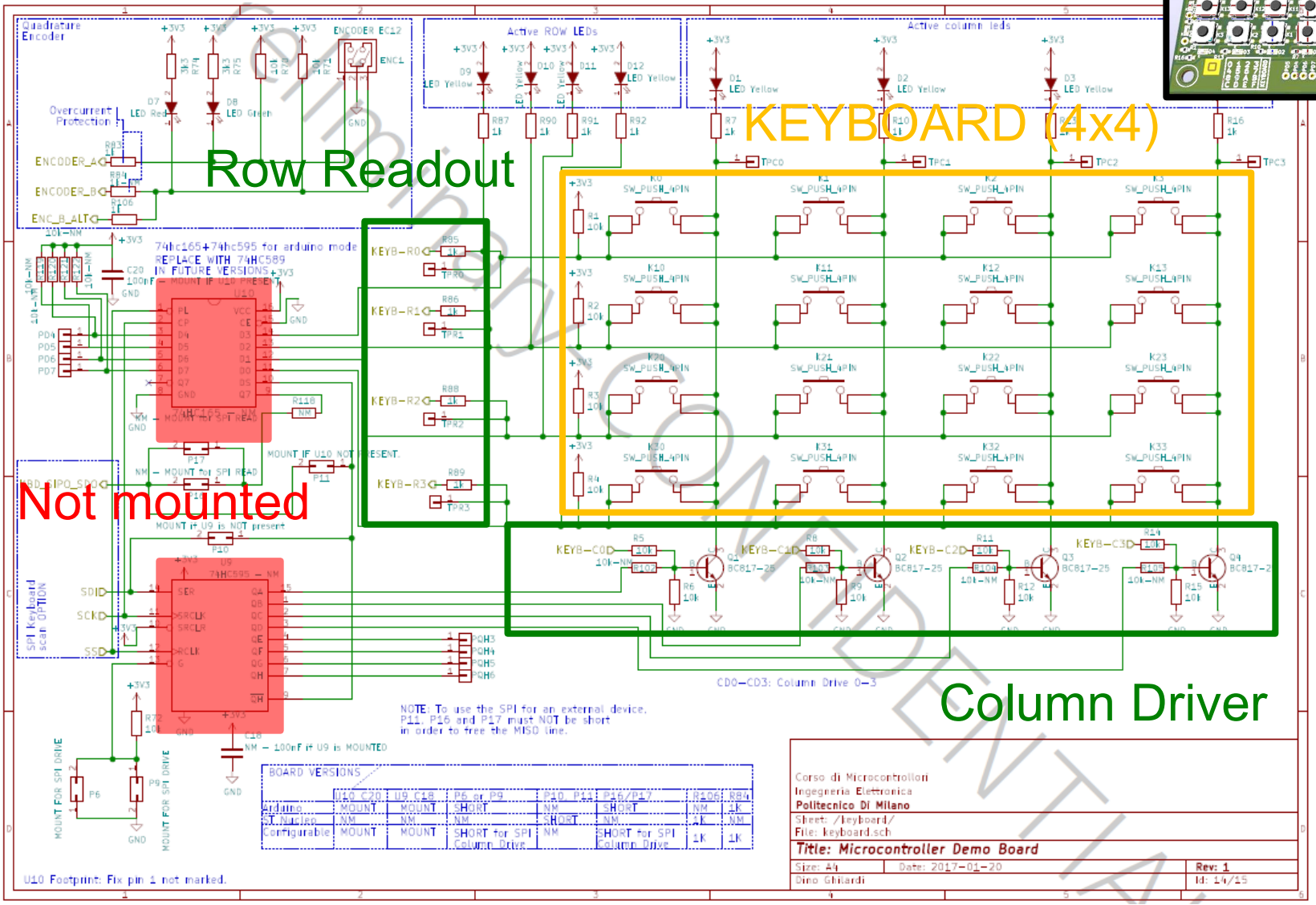


In our PMDB16 board:

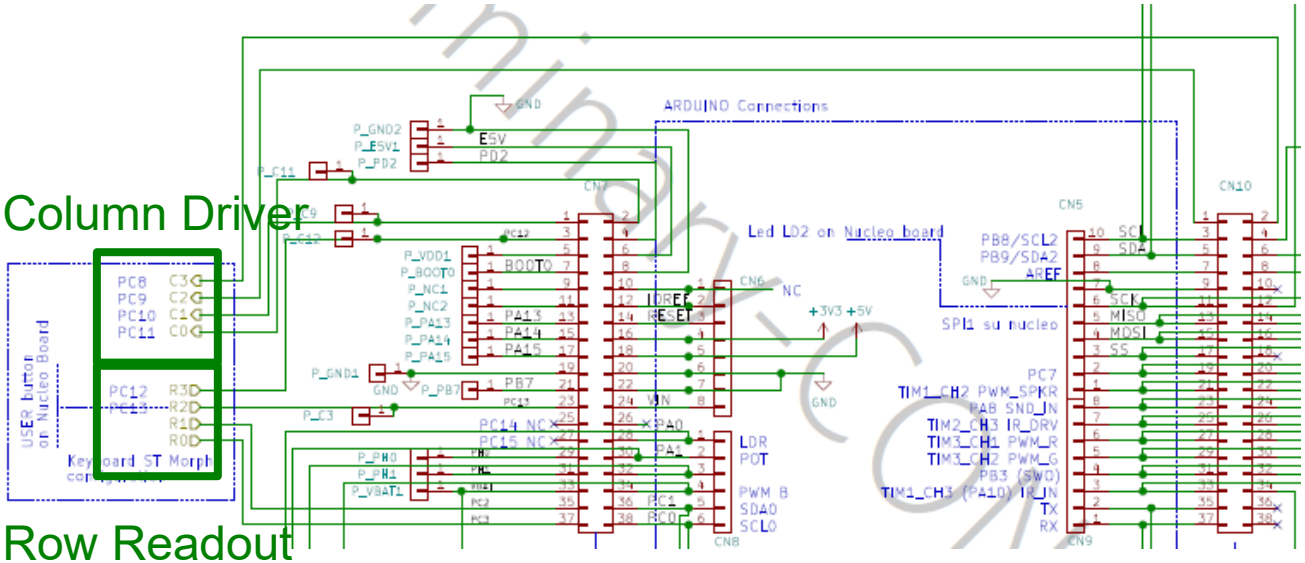
- U9 not mounted, instead we have P10
- U10 not mounted, instead we have P11

KEYBOARD – Schematic overview





KEYBOARD – Pinout configuration

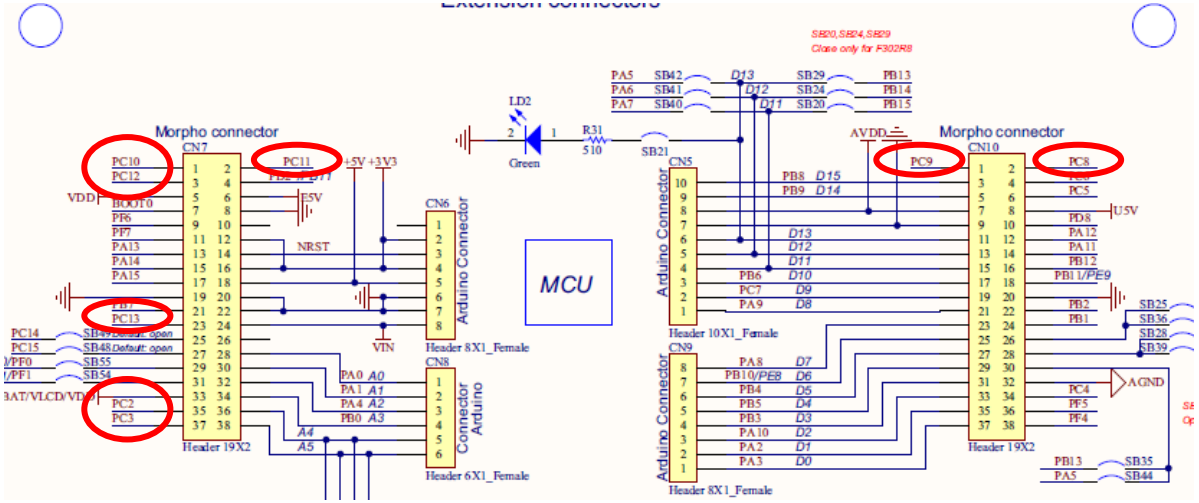


- PC2
- PC3
- PC12
- PC13

GPIO input

- PC8
- PC9
- PC10
- PC11

GPIO output



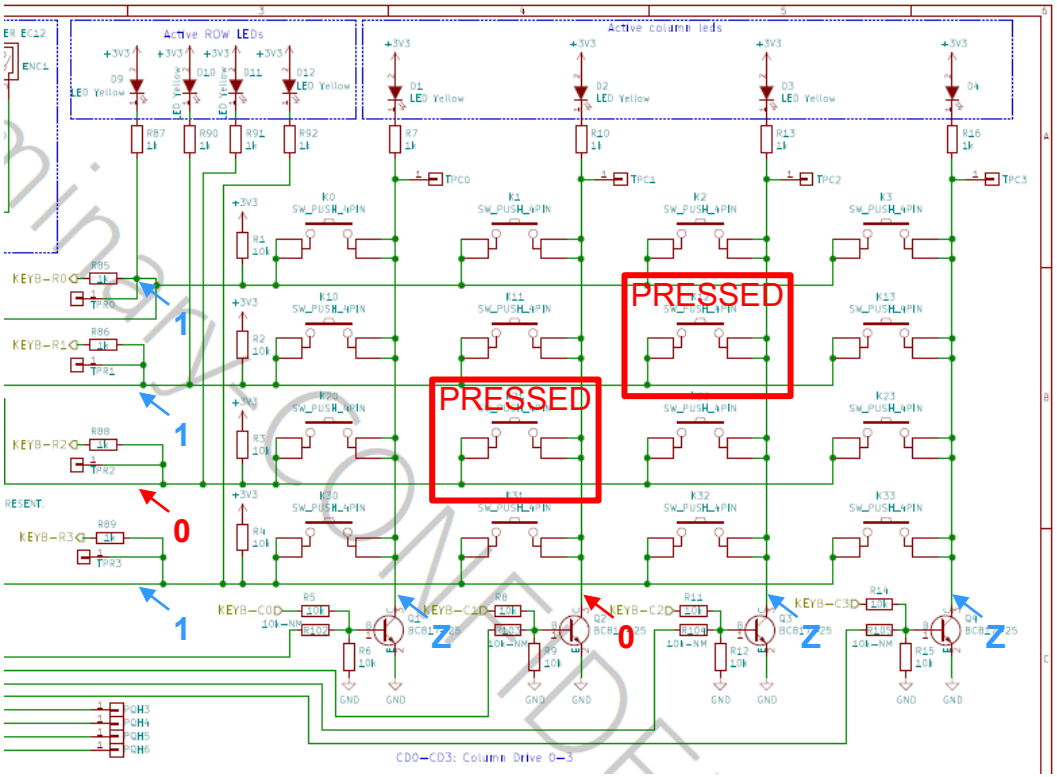
KEYBOARD – How to read a key

Pull-up resistors on the row lines

Open collector transistors on the column line pull it to ground or leave it floating

To read a key:

- **Scan** periodically across the keyboard by enabling **one column at a time**
- **Read** the 4 **rows**:
  - if a row is high (='1') then the key is NOT pressed
  - Else, the key IS pressed
- To **debounce**, wait for a minimum keypress duration before calidating the input



Project 1a - Objective

Objective

Scan each column and read the keyboard

using polling

Project hints

- Configure the 8 GPIOs required to scan the keyboard.
- Within your main loop, select one column at a time, wait for a certain amount of time, then read the status of the 4 rows. Scan through all 4 columns.
- For each pressed key, send the corresponding character to PC via UART. Do not repeat the key send if the key remain pressed at the following iterations. Send the code again if the key is released and pressed again.
- Compile and debug the code. (Suggestion: step through each column slowly and check the code operation)

Project 1b - Objective

Objective

Scan each column and read the keyboard
using a timer interrupt

Project hints

- Configure the 8 GPIOs required to scan the keyboard.
- Setup a timer to provide an interrupt every few ms. In the callback, acquire the status of the 4 rows and then enable a new column. Figure out a way to keep track of when each of the 16 keys is pressed and for how long.
- Within the loop(), if a key is pressed long enough, send the corresponding character to PC via UART. Do not repeat the key send if the key remain pressed. Send the code again if the key is released and pressed again.
- Compile and debug the code. (Suggestion: step through each column slowly and check the code operation)

Project 1 - Setting on CUBE pinout (1)

Starting from a default model project

Select the following GPIO_input and GPIO_output ports