POLITECNICO
DI MILANO

www.polimi.it

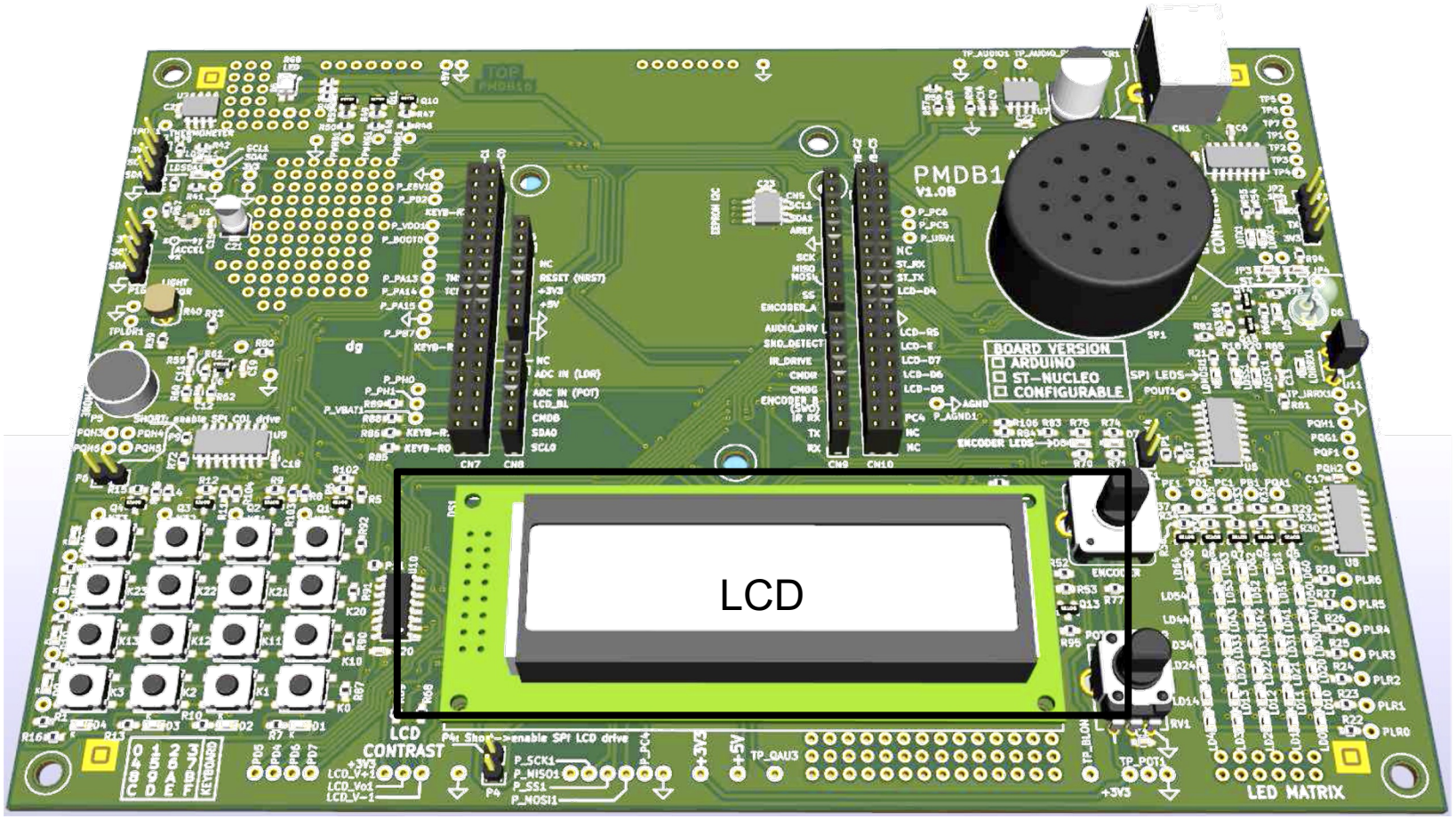




STM32 – LCD

Federica Villa

LCD – *Evaluation Board Components Overview*

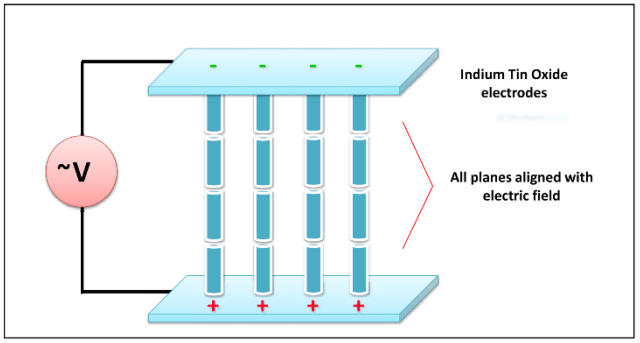


LCD technology

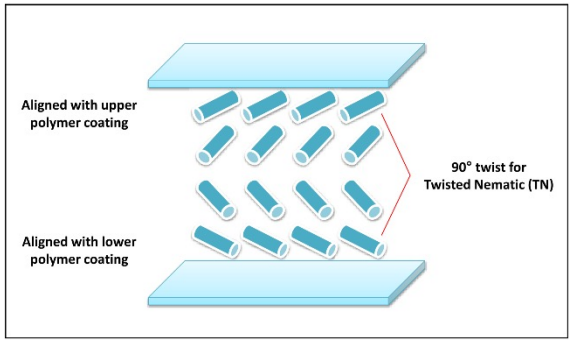
LCD: **L**iquid **C**rystal **D**isplay

- Technology leveraging the nematic phase of liquid crystals
- Liquid crystals in nematic phase flow as a liquid but have the same optical properties as those of crystals
- They can be easily oriented by applying electric/magnetic fields
- A segment of an LCD is considered ON when enough electric potential is applied between the segments and common electrodes
- A segment of an LCD is considered OFF when insufficient electric potential is applied between the segment and common electrodes

ON



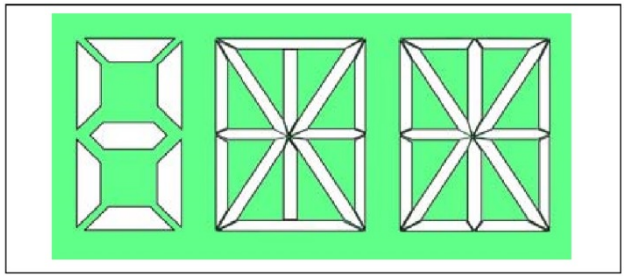
OFF



LCDs classification

Segment Displays

- Usually 7, 14 or 16 segments used to create numbers and letters
- Good contrast and readability in sunlight
- Typical application of segment displays are in calculators, digital clocks and other applications that don't require an high resolution

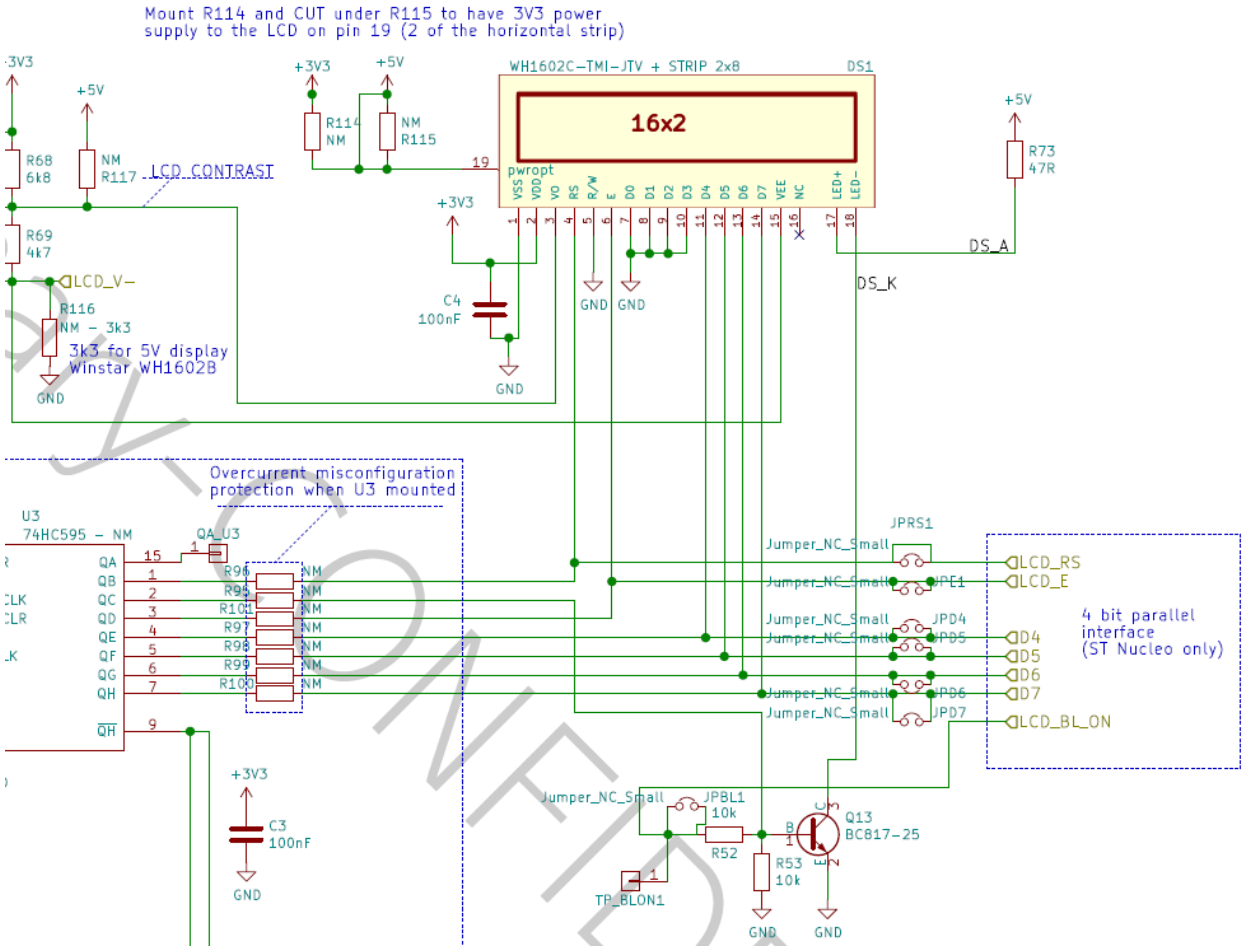


Dot Matrix

- Always multiplex type display, because of the large number of pixel required → pin limitations use a driver.
- Can create more detailed letters and numbers, as well as custom graphic symbols
- Our POLIMI expansion board embeds a 16x2, 5x8 dot matrix LCD display

Step 1

- Let's open the datasheet of the expansion board



Interface between LCD and NUCLEO:

- **LCD_RS**
- **LCD_E**
- **D4**
- **D5**
- **D6**
- **D7**
- **LCD_BL_ON**

Let's identify their purpose

Step 2

- Let's open the datasheet of the LCD module (Winstar WH1602C)

Feature

1.5x8 dots includes cursor
2.Bulit-in controller (ST7066 or Equivalent)
3.5V power supply (Also available for 3V)
4.N.V, optional for 3V power supply
5.1/16 duty cycle
6.LED can be driven by PIN1, PIN2, PIN15, PIN16 or A and K
7.Interface : 6800, option SPI/I2C (RW1063 IC)

Pin No.	Symbol	Description
1	V_{SS}	Ground
2	V_{DD}	Power supply for logic
3	V_O	Contrast Adjustment
4	RS	Data/ Instruction select signal
5	R/W	Read/Write select signal
6	E	Enable signal
7	DB0	Data bus line
8	DB1	Data bus line
9	DB2	Data bus line
10	DB3	Data bus line
11	DB4	Data bus line
12	DB5	Data bus line
13	DB6	Data bus line
14	DB7	Data bus line
15	A	Power supply for B/L +
16	K	Power supply for B/L -

What we can find is:

- The controller IC: **Sitronix ST7066**
- What the lines are:
  - **Control** lines (RS, R/W, E)
  - **Data** lines (DB4 – 7)
  - **Backlight** LED (L+, L-)

Step 3

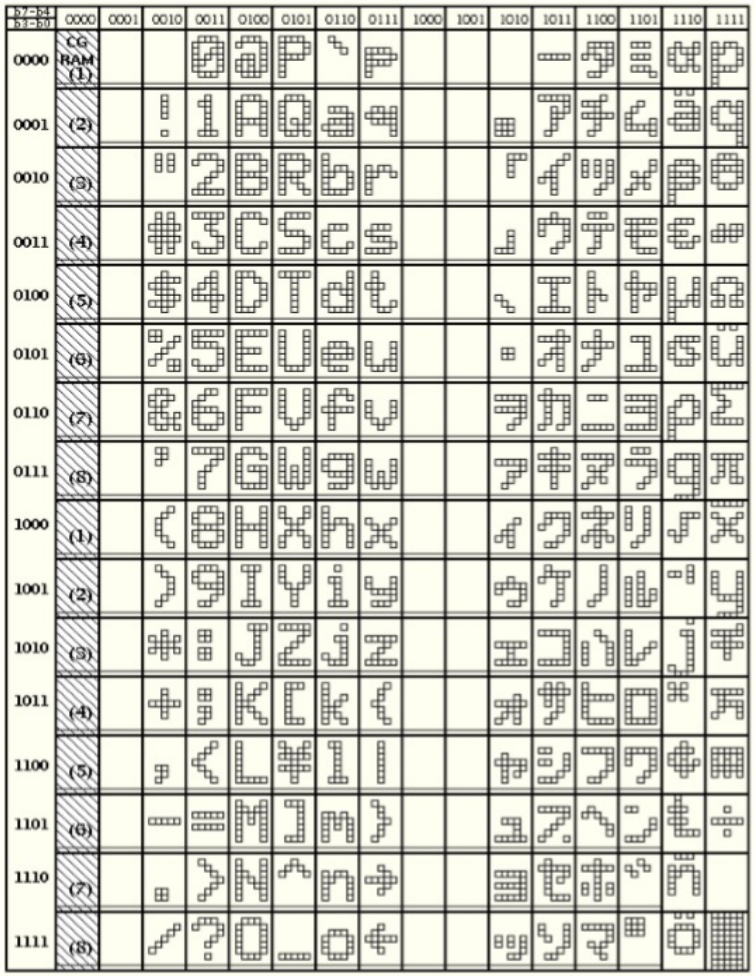
- Let's open the controller datasheet

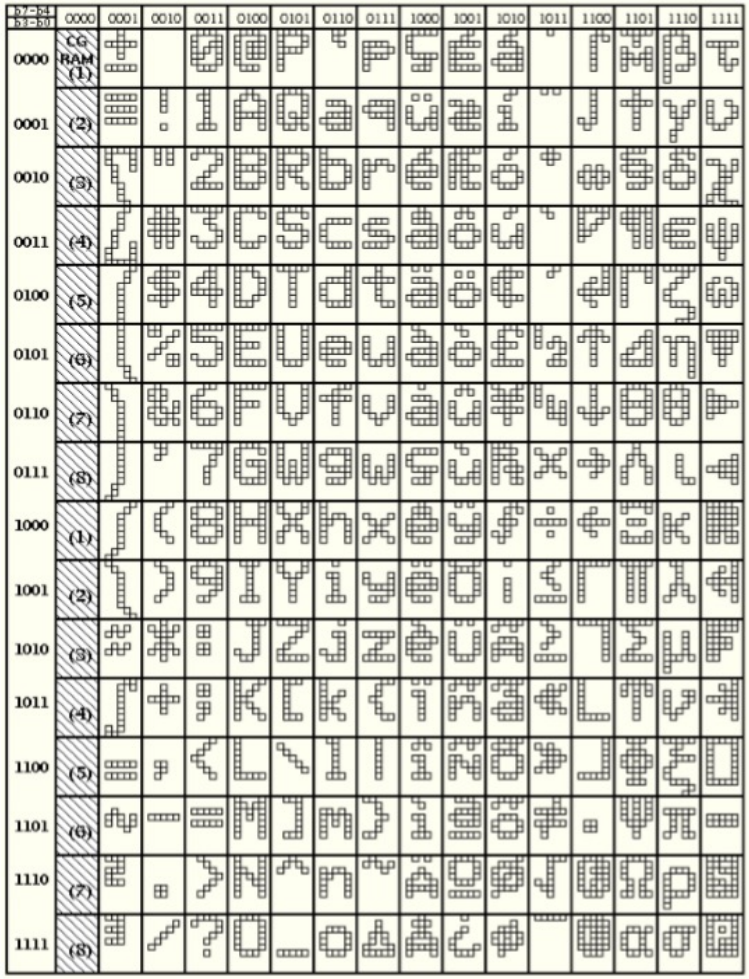
NAME	NUMBER	I/O	INTERFACED WITH	FUNCTION
RS	1	I	MPU	Select registers. 0: Instruction register (for write) Busy flag: address counter (for read) 1: Data register (for write and read)
R/W	1	I	MPU	Select read or write. 0: Write 1: Read
E	1	I	MPU	Starts data read/write.
DB4 to DB7	4	I/O	MPU	Four high order bi-directional tristate data bus pins. Used for data transfer and receive between the MPU and the ST7066. DB7 can be used as a busy flag.
DB0 to DB3	4	I/O	MPU	Four low order bi-directional tristate data bus pins. Used for data transfer and receive between the MPU and the ST7066. These pins are not used during 4-bit operation.
CL1	1	O	Extension driver	Clock to latch serial data D sent to the extension driver
CL2	1	O	Extension driver	Clock to shift serial data D
M	1	O	Extension driver	Switch signal for converting the liquid crystal drive waveform to AC
D	1	O	Extension driver	Character pattern data corresponding to each segment signal
COM1 to COM16	16	O	LCD	Common signals that are not used are changed to non-selection waveform. COM9 to COM16 are non-selection waveforms at 1/8 duty factor and COM12 to COM16 are non-selection waveforms at 1/11 duty factor.

Step 3: Main internal registers - 1

CGROM (Character Generator ROM):

contains the binary values required to display a preset of characters





Step 3: Main internal registers - 2

CGRAM (**C**haracter **G**enerator **RAM**):

contains space for up to 8 custom characters 5x8 dot matrix that can be programmed (volatile memory, content is lost if power is disconnected)

Step 3: Main internal registers - 3

DDRAM (**D**isplay **RAM**):

stores display data as 8-bit codes (address of the character in CGROM/RAM).

Display Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
DDRAM Address	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
For Shift Left	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10
	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50
For Shift Right	27	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E
	67	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E

Step 4 – Instruction Table

To write a character

- Set DDRAM address to the position where we want to draw it

To write a second character, if on the following position, just write the 8b data: address is auto-incremented.

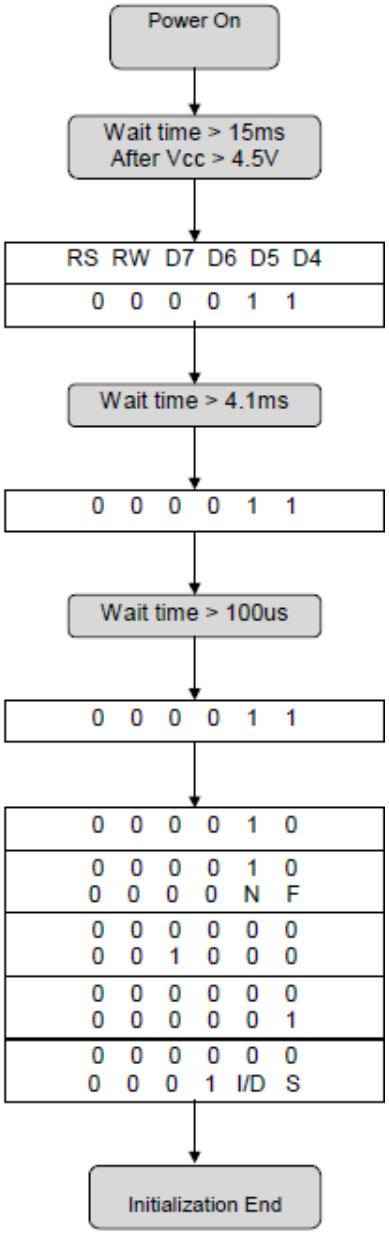
- Write 8b CGROM address of the selected character

CGROM address corresponds to the char ANSI code.

Instruction Table:

Instruction	Instruction Code										Description	Description Time (270KHZ)
	RS	RW	DB 7	DB 6	DB 5	DB 4	DB 3	DB 2	DB 1	DB 0		
Clear Display	0	0	0	0	0	0	0	0	0	1	Write "20H" to DDRAM. and set DDRAM address to "00H" from AC	1.52 ms
Return Home	0	0	0	0	0	0	0	0	1	x	Set DDRAM address to "00H" from AC and return cursor to its original position if shifted. The contents of DDRAM are not changed.	1.52 ms
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 us
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1: entire display on C=1: cursor on B=1: cursor position on	37 us
Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x	Set cursor moving and display shift control bit, and the direction, without changing DDRAM data.	37 us
Function Set	0	0	0	0	1	DL	N	F	x	x	DL: interface data is 8/4 bits NL: number of line is 2/1 F: font size is 5x11/5x8	37 us
Set CGRAM address	0	0	0	1	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Set CGRAM address in address counter	37 us
Set DDRAM address	0	0	1	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Set DDRAM address in address counter	37 us
Read Busy flag and address	0	1	BF	AC 6	AC 5	AC 4	AC 3	AC 2	AC 1	AC 0	Whether during internal operation or not can be known by reading BF. The contents of address counter can also be read.	0 us
Write data to RAM	1	0	D7	D6	D5	D4	D3	D2	D1	D0	Write data into internal RAM (DDRAM/CGRAM)	43 us
Read data from RAM	1	1	D7	D6	D5	D4	D3	D2	D1	D0	Read data from internal RAM (DDRAM/CGRAM)	43 us

Step 5 – Initialization (4 bit mode) - 1



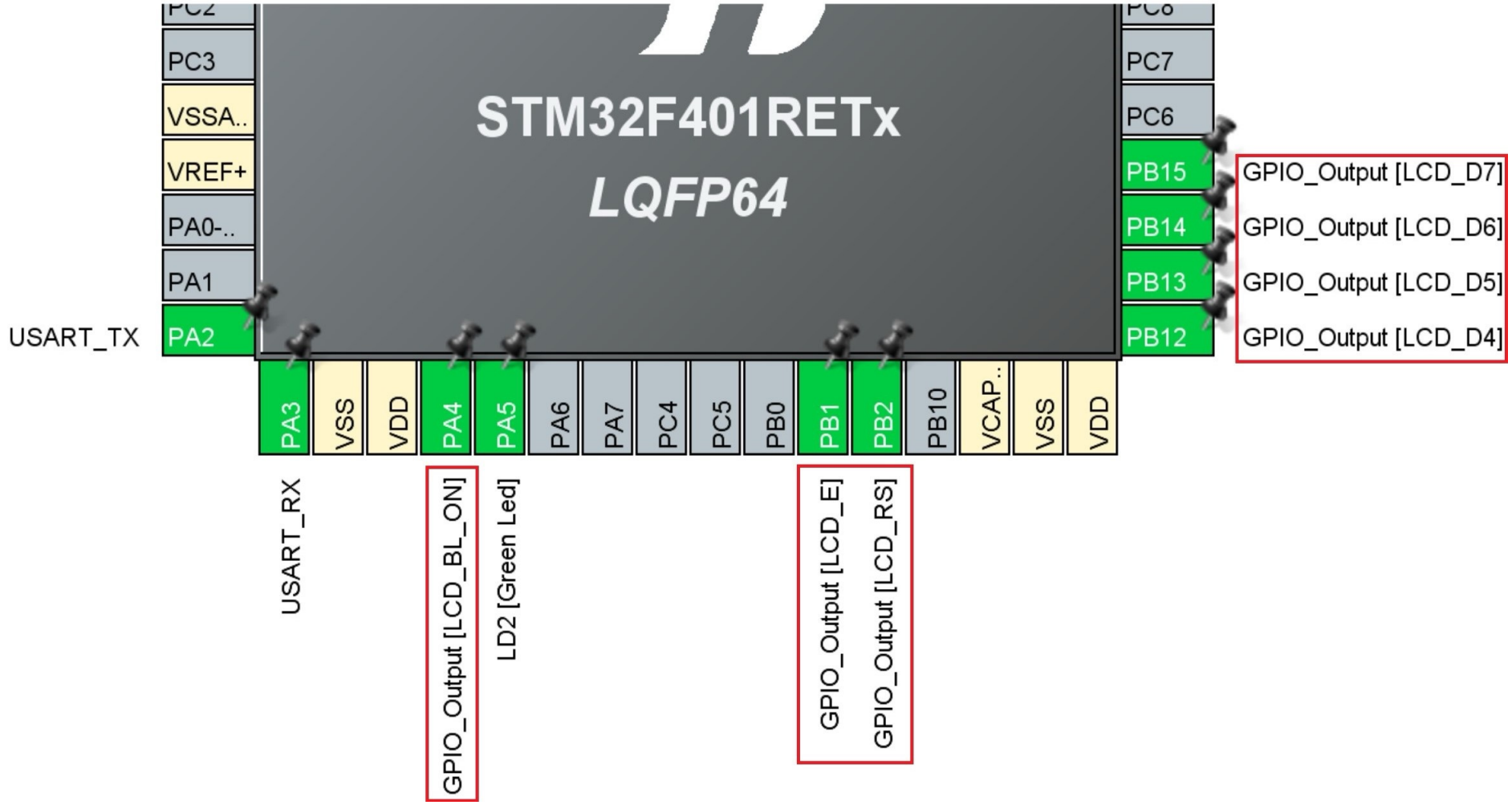
To write anything:

- Set RS; RW; D7-4 values to the desired value
- Give a pulse 0 →1→ 0 to the E line to latch the value in the controller (Page 23)

After initialization, to write a byte:

- First write the most significative nibble (00110001)
- Then the least significative nibble (00110001)

Pinout Configuration



LCD library

To use the LCD:

- Configure the required GPIO pins (see previous slide) as outputs

- Import the “PMDB16_LCD” library in your STM32cubeIDE project:
  - Copy the “PMDB16_LCD.c” file to your project/Core/Src folder
  - Copy the “PMDB16_LCD.h” file to your project/Core/Inc folder
  - Add `#include "PMDB16_LCD.h“` in your main.c

- Initialize the LCD controller before the `while (1)` loop.
- Write to the LCD with the provided functions.

See next slide for an overview of the functions.

LCD library high level functions

`void lcd_initialize();` Initializes the LCD controller.

`void lcd_backlight_ON();` Turns ON(OFF) the LCD backlight.
`void lcd_backlight_OFF();`

```
void lcd_println(char string[], uint8_t row);
```
Prints the string on the top (0) or bottom (1) row of the LCD.
Maximum string length = 16 characters.

```
void lcd_drawBar(int value);
```
Prints a bargraph on the bottom row of the LCD controller. Value range: 0 to 80. Each increment corresponds to enabling one extra column of the dot matrix display.

```
void lcd_clear();
```
Clears the entire display.