Screen

⌐→ screen tyle

    ↳ LCD = character Display

    ↳ Dot Matrix

    ↳ graphical Display ↳ Colored Display.

↳ [LCD] → liquid crystal Display

         ↳ size 16 x 2

      ↳ Pin out

      ↳ Block Disram + Component

      ↳ work Flow

            ↳ How Communicate

            ↳ Commonds

Display tyle

↳ led

    ↳ segment

      ↳ screen

Pinout

① Power

    ① gnd → Vss → ov

    ② Vcc → VDD → 5v

    ③ V_E → Pot

GND
Vcc
VEE
RS
R/W
EN
DB0
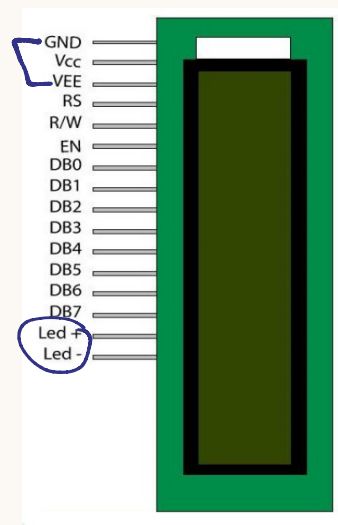DB1
DB2
DB3
DB4
DB5
DB6
DB7
Led +
Led -

② Control

    ① Rs → Register select

        ↳ Data For Display

        ↳ Command For lcd

    ② R/w → Read /write

③ E → EN → Enable
                    ↳ Enable High   → Enable low

Pulse
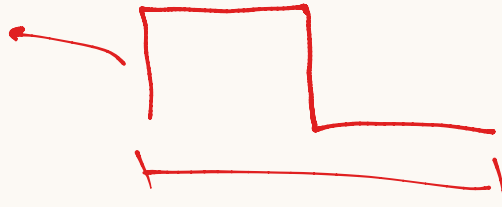
③ <u>Data Pin</u> :. Data / Command
* D_0 ⟶ D_7 ⟹ 8 bit → <u>8</u> bit Mode
* D_4 ⟶ D_7 ⟹ 4 bit → <u>4</u> bit Mode

④ Back source light
        led + ⟶ 5v
        led - ⟶ 0v

<u>LCD Block Disgram</u> → Character → AS$C\underline{II}$
                                         I ⟶ ' \\

Pinout ⟶ [LCD Panel / LCD Controller] ⟶ Proccessor
         [CGROM | CGRAM | DDRam]
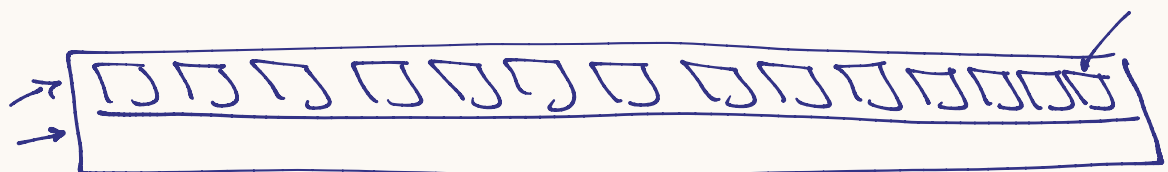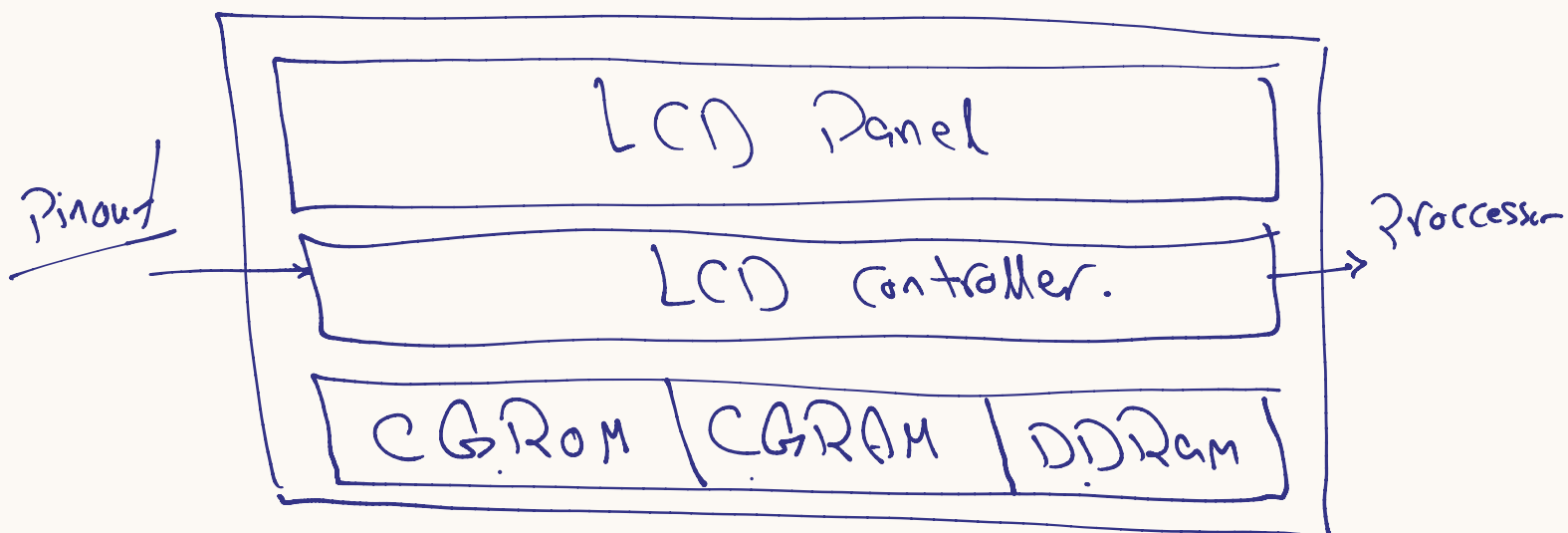
16×2 ⟹ 16 ⟶ No of display digit
                    2 ⟶ No of line

DD Rom

↳ Display Data Ram (volatile memory)

↳ store for date represent in 8 bit character

↳ Start Address ⟶ 0X00
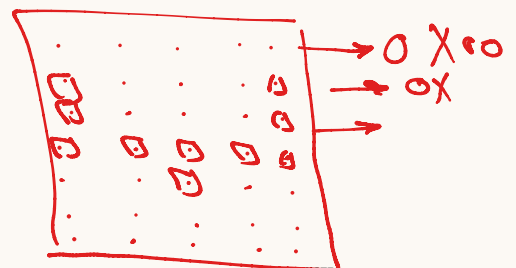
| 0X00 | | 0X0F |
| 0X40 | | 0X4F |

⟶ CGRom :. Character Gerenating Rom.

↳ save a petterns for all Characters
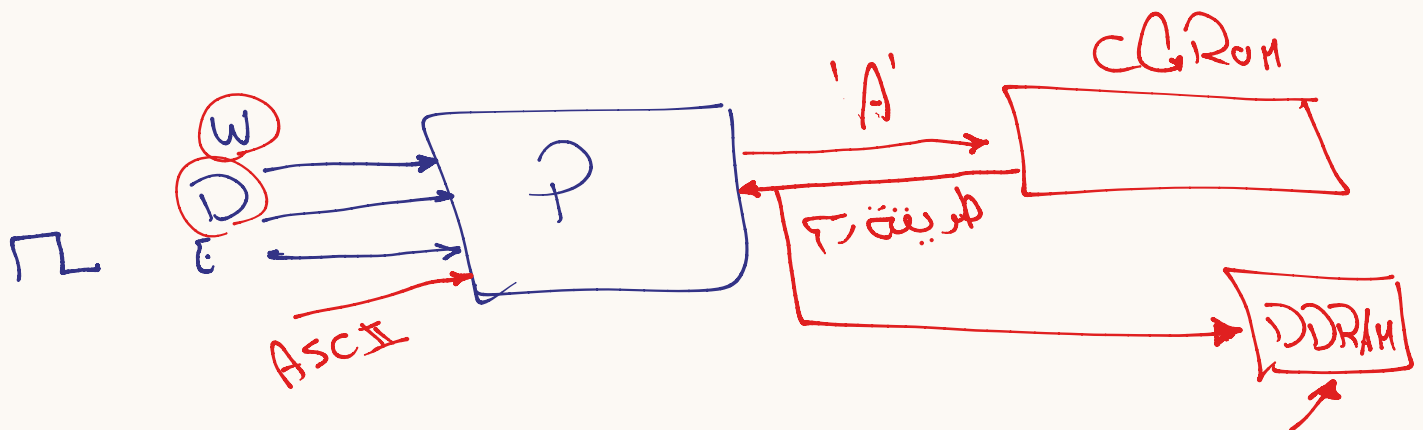
CGRAM ⟶ ROM Memory store New character

↳ Must sent Draw way / ؟ ـب

Must sent Draw way for char

Work Flow

└ **Need to send Command From MCU**

① ⟶ Rs ⟶ Low ⟶ 0 Command

② ⟶ Rw ⟶ low ⟶ 0 Write

③ ⟶ E ⟶ High ⟶ 1 ⌐

④ ⟶ D_0 ⟶ D_7 (Command value)

⑤ ⟶ E ⟶ low ⟶ 0 ⌐

⟶ step if You will sent Command by 8 bit Mode

Rs
└ 0 ⟶ Com
└ 1 ⟶ data

Rw
└ 0 ⟶ write
└ 1 ⟶ Read

└ **Need to sent Data From MCU** (8bit - Mode)

① Rs ⟶ High ⟶ 1

② Rw ⟶ low ⟶ 0

③ E ⟶ High ⟶ ⌐

④ Data value ⟶ D_0 ⟶ D_7

⑤ E ⟶ Low.