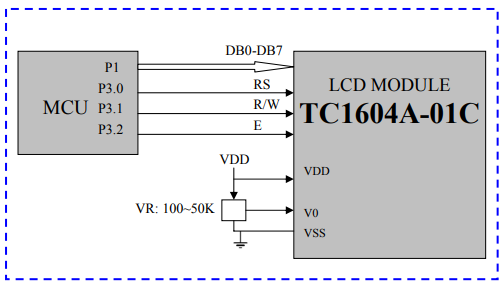
Currently making a plan to work on LCD driver

<https://leeselectronic.com/en/product/5576-lcd-display-4x16-char-tc1604a-01.html>

Data from datasheet

<https://dosya.motorobit.com/pdf/TC1604a.PDF>



Pin setup

**VCC,V0,VSS** - controls brightness, can keep at 3.3V can dim by attaching potent or resistor

**DB0-DB7:** – 8 GPIO’s needed

**RS – 1:** Data register for read and write, 0: Instruction register (for write)

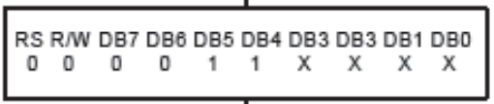
**R/W - 1:** (Read mode, 0 Write mode)

### **Initialization sequence**

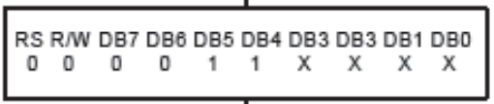
Power on

If VDD>4.5 wait 15ms

Send for 8-bit length interface



Wait 100us and send again



AFTERWARDS

Display Off - 0 0 0 0 0 1 0 0

Display Clear: 0 0 0 0 0 0 0 1

Entry Mode Set: 0 0 0 0 0 0 1 I/D S

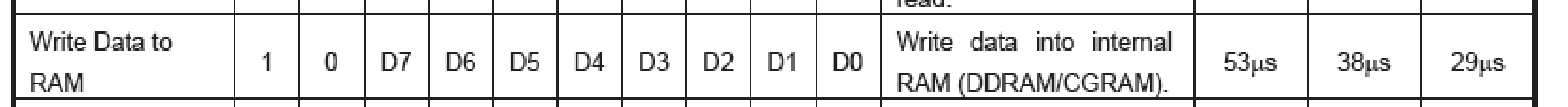
I/D = 1 → Cursor moves right after each character.

I/D = 0 → Cursor moves left after each character.

S = 1 → Display shifts instead of the cursor.

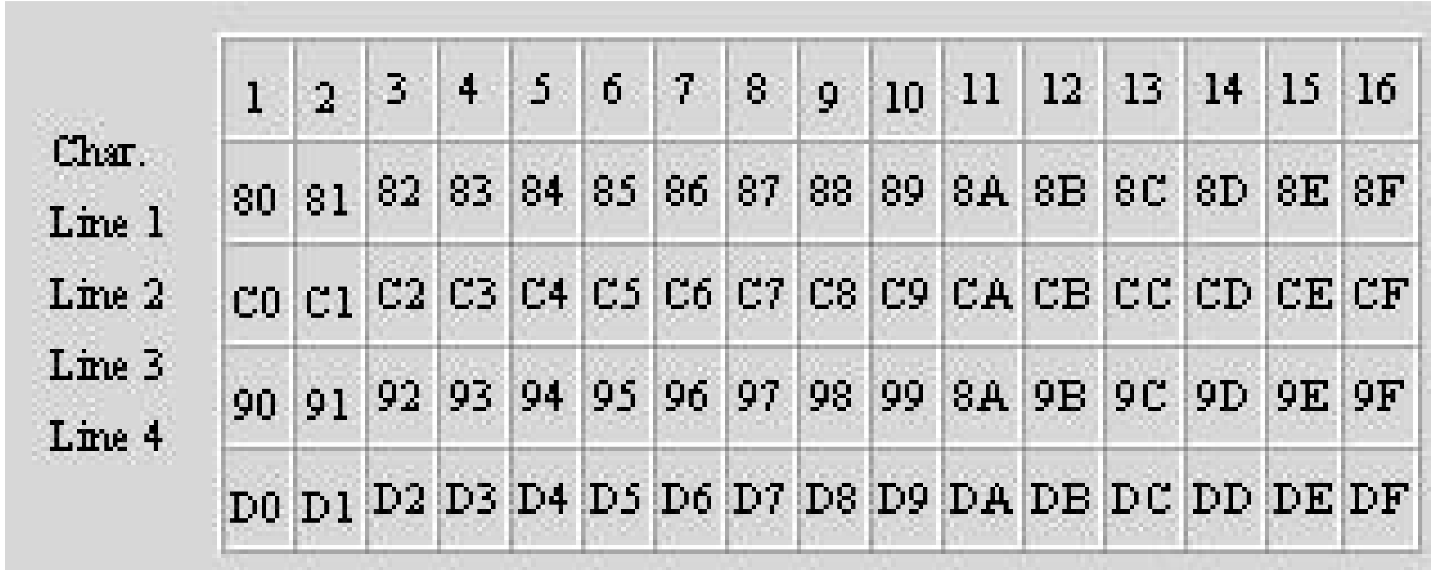
S = 0 → Display does not shift.

### **Sending characters:**



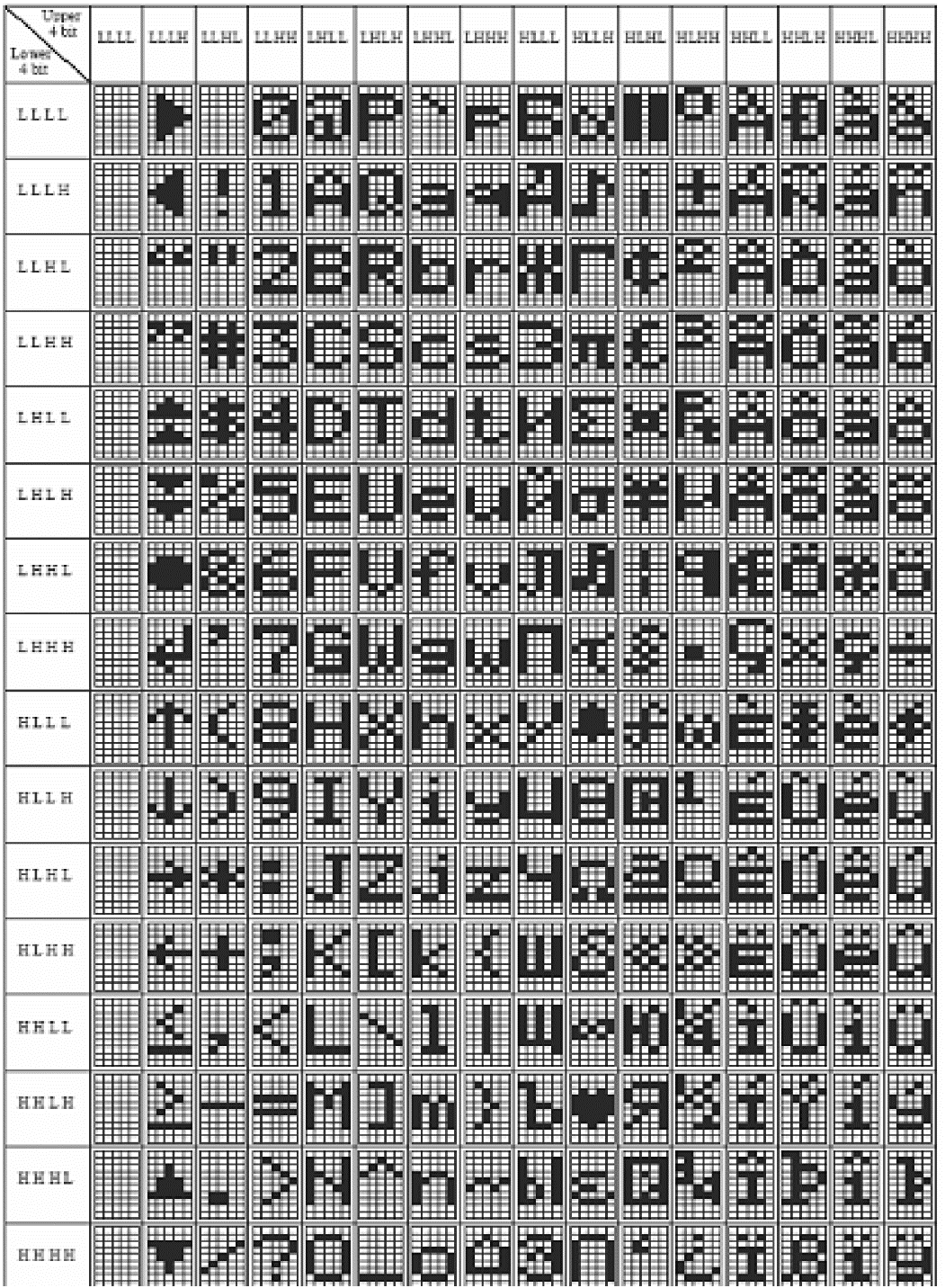
It should automatically shift DRAM data

# DRAM DATA MEMORY address



# CHARACTER GENERATION

EX) LLLLHLH -->0000 0101 --> corresponds to the letter P



Creating temporary module.

**Timing calculations**

MINIMUM TIMING is 100us,

I will choose , which gives 9765.625Hz or = 102.4us

Power on wait time atleast 15ms.

**Initialization sequence**

Pinouts:

## E pin – assumption: All commands are registered through e trigger. So we can actually put that as the clock.

## R/W pin –

**Read mode**(lcd -> MCU - 1)

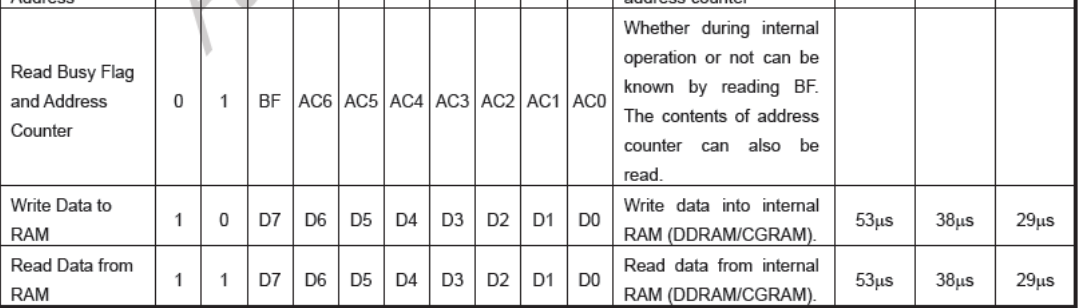
**write mode (mcu -> LCD - 0) – don’t we always keep this on?**

## RS pin:( selects registers)

**1 – data register <- I believe we are doing this**

0 - instruction register or busy-flag address counter

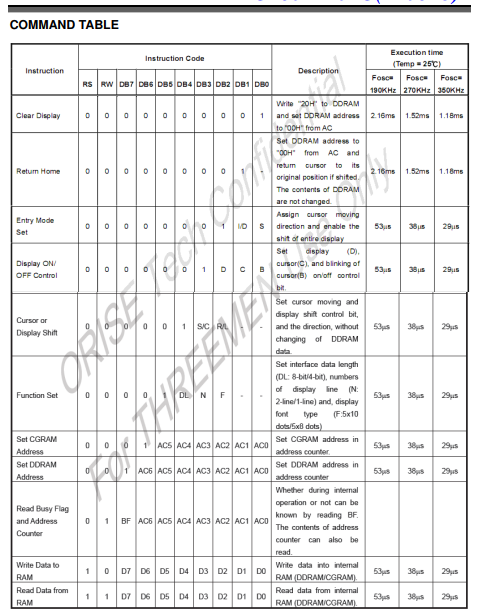
Keep both these registers low, except for the following commands.



Afterwards we are going to be writing to MCU R/W -> 0

RS-PIN UNKNOWN what is data-register, vs instruction register

**Instruction register**



Data register is also known as RAM. Hard to know, I can try both methods.

But for now RS-> 1 for sending data.



Figure 1 Location of writing characters

