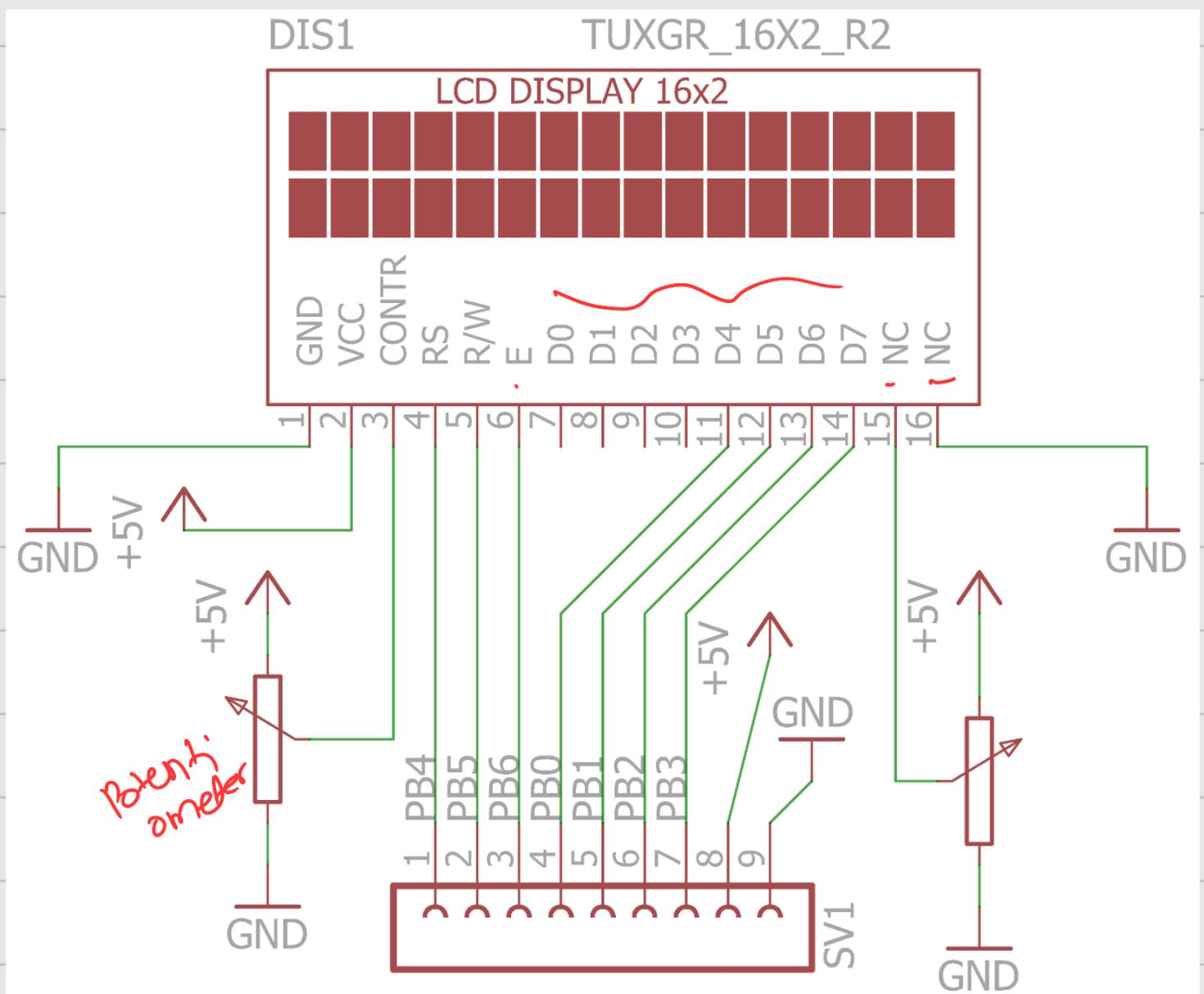


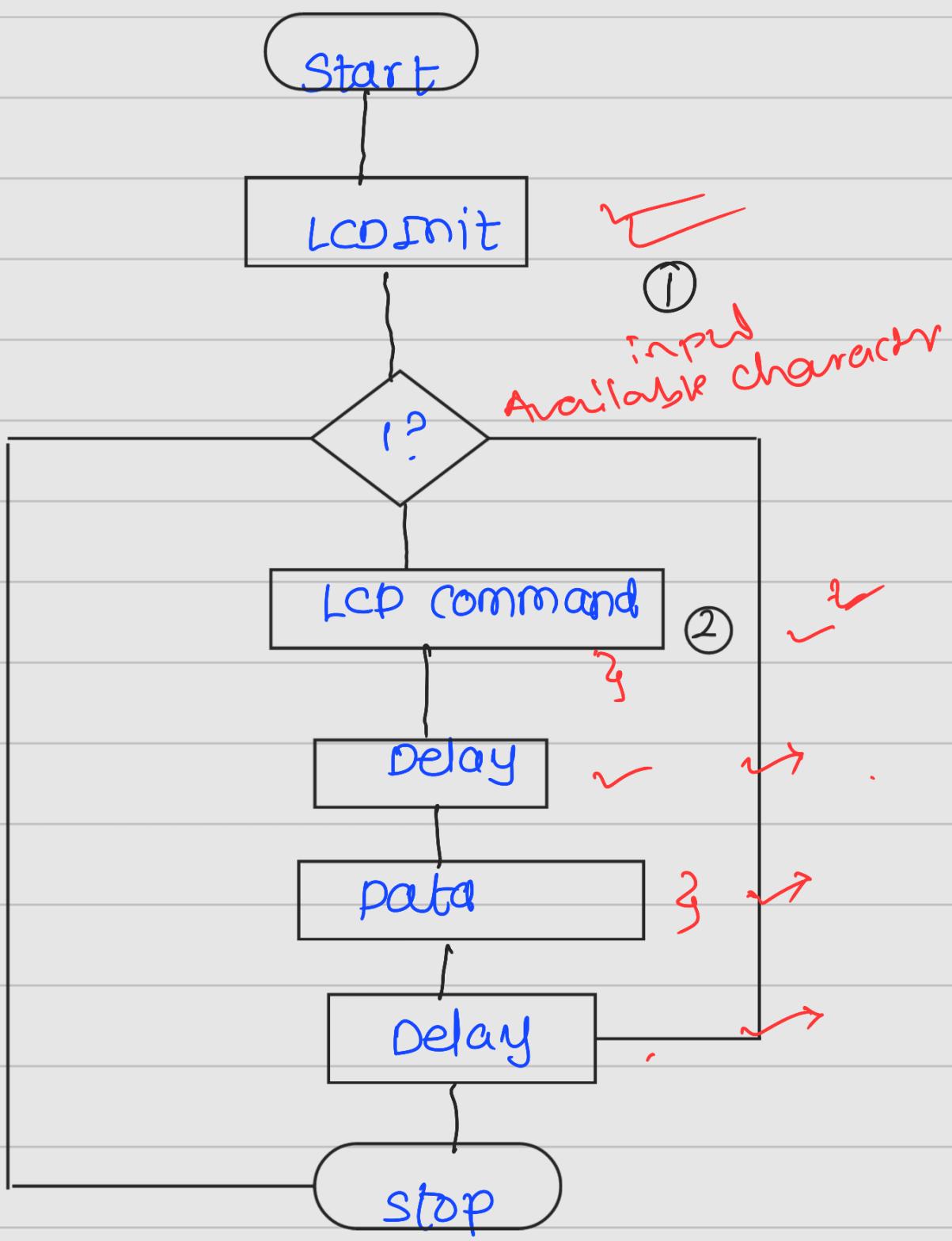
LCD - 16x2

- HD44780 based display
- will print the integer, character, float as well as custom characters.



## ① Connection :

Pin.no	Name	Connected to Tiva board Pin
1	VSS	→ Gnd
2	VDD	→  VBUS -5V
3	V <sub>o</sub>	→ Pin 2 . of Potentiometer
4	RS	→ PA5 ✓
5	R/W	→ PA6 ✓
6	E	→ PA7 ✓
7	D <sub>0</sub>	- PDO
8	D <sub>1</sub>	PDI
9	D <sub>2</sub>	PD2
10	D <sub>3</sub>	PD3
11	D <sub>4</sub>	PC4
12	D <sub>5</sub>	PC5
13	D <sub>6</sub>	PC6
14	D <sub>7</sub>	PC7
15	A	backlight +V -5V
16	K	Gnd



## ⑧ Controlling LCD

Command register ✓

RS = 0      RS = 0  
write 0x01  
Send command (0x01)

Data Register ✓

RS = 1  
content 1

① 0X01 - Clear LCD ✓

② 0X02 - Return Cursor Home ✓

③ 0X06 - Shift cursor Right

④ 0X0F - Turn Display ON

⑤ 0X80 - Line 1 Begin ↗  
                            ↘ 0x80  
                            ↗ 0xC0 16x?

⑥ 0x C0 - Line 2 Begin

⑦ 0x 38 - 8-bit Data ✓ ↗

⑧ 0x28 - 4-bit Data ✓ ↗

ASCII value :

A, C, #

\* Why delay between commands ??

LCD controller design for low power application. Controllers

delay() ✓

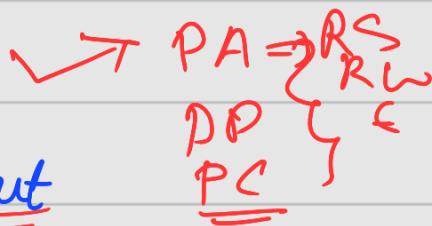
\* Enable Pin ?? ✓

P101      n

Com · Secure ⇒ enable 1  
             command

secure data ⇒

i) LCD\_Init() :-



- Enable clock to GPIO PORT
- Set GPIO Pins as GPIO output
- R/W = 0 → R/W = 0
- function set command → 0x38 = 8 bit, 2 display lines  
5x7

Entry mode set comm. → 0x06 ⇒ auto increment

Display control → 0x0F. ✓

Display control → 0x01 ✓

ii) LCD Command & Data function

void LCD\_Command ( unsigned char cmd )

{  
  (a) RS = 0 ✓      RS = 0

  (b) wait command ✓      0x06

  (c) secure command ... enable pin: E ↑

  (d) RS=0 set back ✓

  (e) delay ✓

}

LCD\_Command

LCD\_sendData ("A")

Void LCD\_Data (consigned char data)

{  
    <a> RS=1 ; ✓ RS=1  
    <b> write data ; .wdata  
    <c> Secure data      spec  
    <d> RS=0  
    <e> delay  
}

\* Custom character on LCD :

- Requires a knowledge of CG-RAM & LCD chip
- get controller to build
- HD4478 controller - build by Hitachi

CG-RAM :

- CG = Custom Generated
- store our custom character once we declare them in our code
- CG-RAM = Size - 64bytes  
= 8 characters at a time ✓

0x40

→ Starting address = 0x40 (in hexadecimal)

Each character requires 8bytes ✓

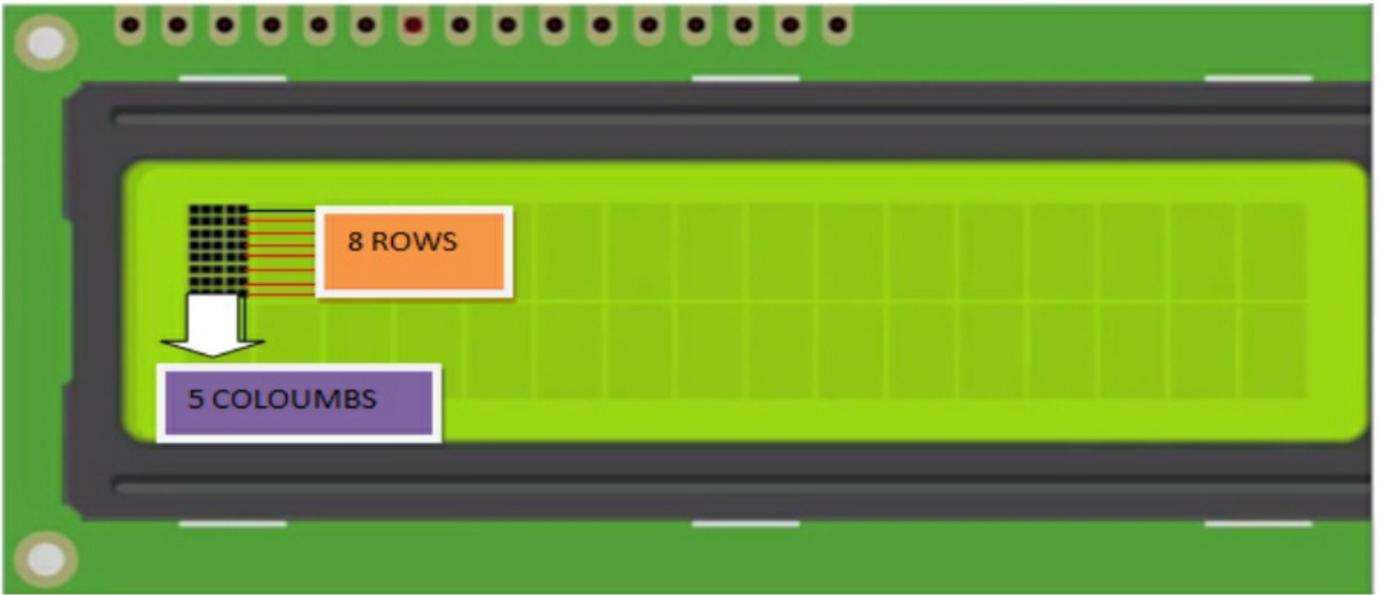
Each pattern is of 1 byte (8bit) ~~8 bytes~~

CG-RAM Characters	CG-RAM Address (Hexadecimal)	Size
www.microcontroller-project.com		
1 <sup>st</sup> Character	0x40	A
2 <sup>nd</sup> Character	0x48	B
3 <sup>rd</sup> Character	0x50	C
4 <sup>th</sup> Character	0x58	Z
5 <sup>th</sup> Character	0x60	
6 <sup>th</sup> Character	0x68	
7 <sup>th</sup> Character	0x70	
8 <sup>th</sup> Character	0x78	

www.microcontroller-project.com

16x2





Each Location =  $5 \times 8$   
 = 8 patterns for 8 rows  
 = 1 byte for each pattern  
 = 8 bytes for one character

$$0x40 = \overbrace{0x40}^{\text{C}} = \{ 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10, 0x10 \}$$

CG-RAM ADRESS	Byte Adress	C1	C2	C3	C4	C5	Pattern for b is
Starting Adress of CG-RAM.		0	0	0	0	0	Row1 = 0x10 = 00010000
0 1 0 0 0 0 0 0	0	1	0	0	0	0	Row2 = 0x10 = 00010000
	1	1	0	0	0	0	Row3 = 0x16 = 00010110
	1	1	0	0	0	1	Row4 = 0x19 = 00011001
	1	0	0	0	1	0	Row5 = 0x11 = 00010001
	1	0	0	0	0	1	Row6 = 0x11 = 00010001
	1	1	1	1	0	0	Row7 = 0x1E = 00011110
	0	0	0	0	0	0	Cursor position

Annotations:

- Red arrows point to the first three bits of the address (0x40) and the first three columns of the memory grid, labeled "First three bits are treated as Don't cares."
- A red bracket groups the last five bits of the address (0x40) and the last five columns of the memory grid, labeled "Rest five bits are loaded at the address".
- A red arrow points to the cursor position at Row7, Column0.
- Red numbers 0x10, 0x16, 0x19, 0x11, 0x11, 0x1E are shown next to their respective row patterns.

① LCD send Cmd ( 0x40 ); ~~RAM~~

→ LCD send Data ( Row1 )

→ LCD send Data ( Row2 )

⋮

→ LCD send Data ( Row7 )

} For one character

A =

① LCDsendCmd (0x48); } BT  
→ LCDsendData (Row1) 0x18 } For 2<sup>nd</sup>  
→ LCD send Data (Row2) 0 } character  
→ LCD send Data (Row3)  
⋮  
→ LCDsendData (Row8) }

0x40

Command	Code									Description	Execution Time	
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Set CG RAM Address	0	0	0	1	0	0	A <sub>CG</sub>	0	0	0	Sets the CG RAM address. CG RAM data can be read or altered after making this setting.	40µs

→ 0x40 → 40

Memory Map

0x40 → Options

Pattern No.	CGRAM Address (A <sub>CG</sub> )	Data rows
1	0x00 - 0x07	<del>DATA</del> Data rows
2	0x08 - 0x0F	
3	0x10 - 0x17	
4	0x18 - 0x1F	
5	0x20 - 0x27	
6	0x28 - 0x2F	
7	0x30 - 0x37	
8	0x38 - 0x3F	

Custom Pattern	Decimal	Hex
	Row 1: 4	0x04
	Row 2: 14	0x0E
	Row 3: 14	0x0E
	Row 4: 14	0x0E
	Row 5: 31	0x1F
	Row 6: 0	0x00
	Row 7: 4	0x04

