LCD → Character Generator RAM / Custom Characters

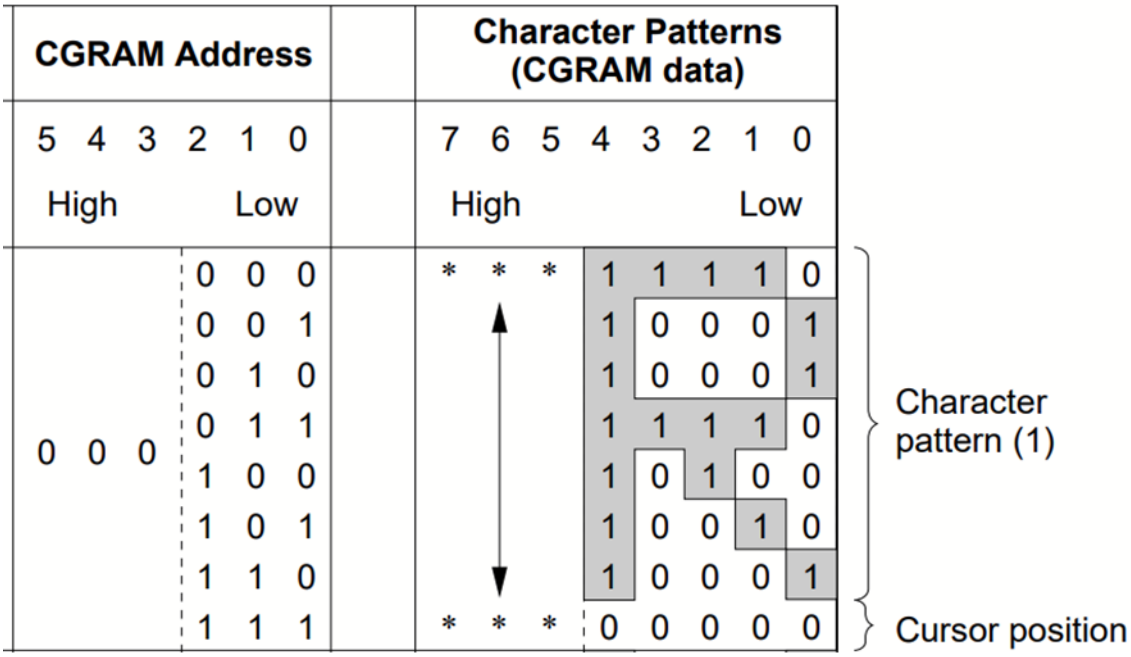
Your LCD has memory space set aside to create/use custom characters.

You may program the pixel patterns for these custom characters.

On your device, you may create eight different custom characters.

Characters are mapped as 5×7 dots, but do include 5×8 where the last row is normally reserved for the cursor.

When programming the pattern, we use one byte per row, but the upper three bits are ignored:



CGRAM Address						Character Patterns (CGRAM data)								
5	4	3	2	1	0	7	6	5	4	3	2	1	0	
High					Low	High							Low	
0	0	0	0	0	0	*	*	*	1	1	1	1	0	Character pattern (1)
			0	0	1				1	0	0	0	1	
			0	1	0				1	0	0	0	1	
			0	1	1				1	1	1	1	0	
			1	0	0				1	0	1	0	0	
			1	0	1				1	0	0	1	0	
			1	1	0				1	0	0	0	1	
			1	1	1	*	*	*	0	0	0	0	0	Cursor position

You enter character generator mode by sending the "Set CGRAM Address" command. After this command, data will be used to program pixel patterns.

Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 µs
			7	6	5	4	3	2	1	0		

COMMAND BYTE

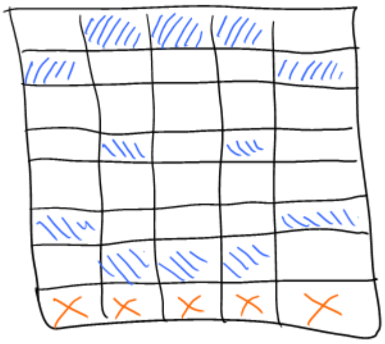
Each custom character starts @ an eight byte offset, so your program function should expect at least 8 bytes of data, and the character to program.

```
// character pos in cgAddr, 8 bytes of character data at cgData
void lcd_CGChar (unsigned char cgAddr, unsigned char const * const cgData)
```

∅→7

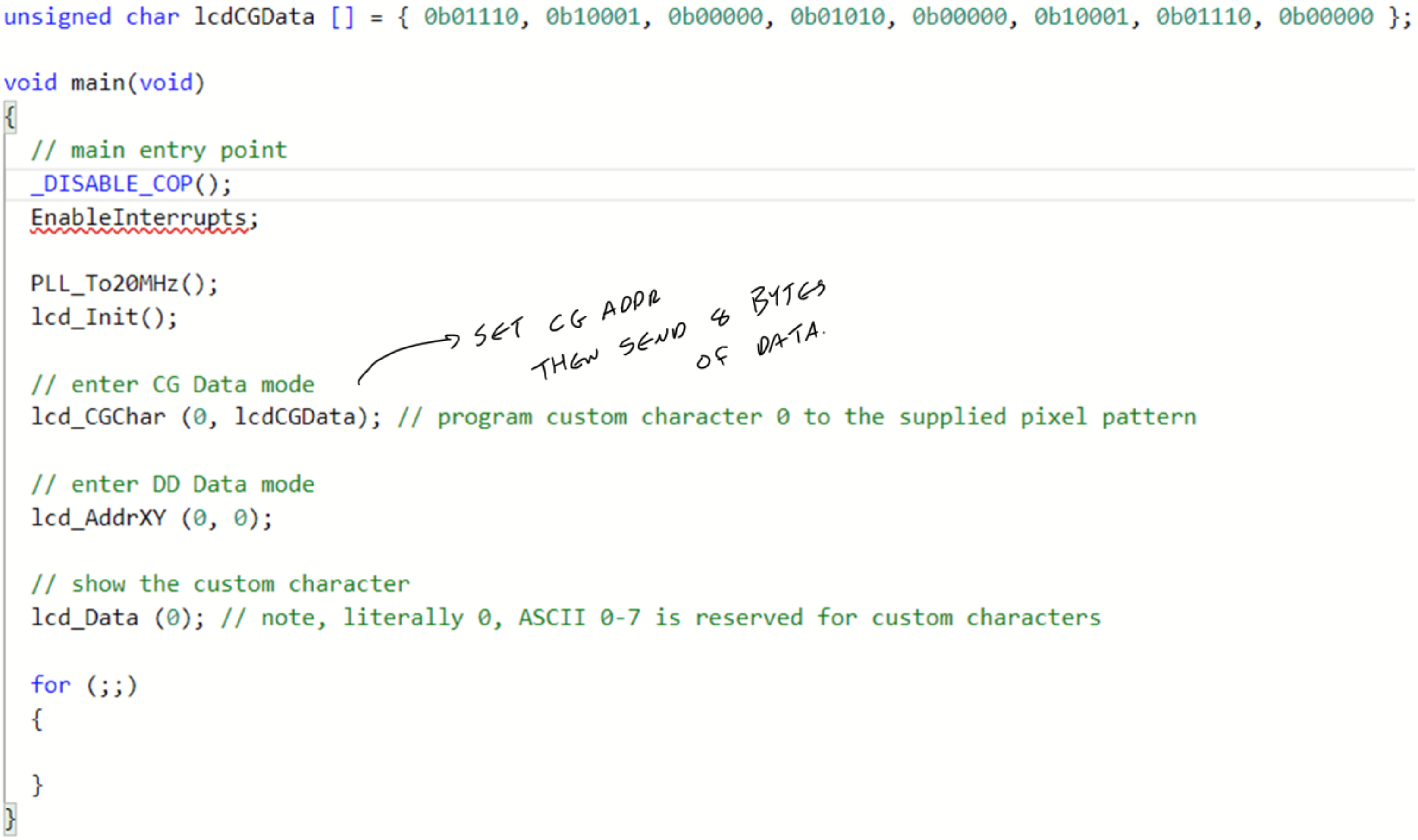
ARRAY OF 8 BYTES w/ pixel pattern.

You would normally use graph paper to draw the pattern, and then determine the byte values per row:



0b000	01110	UPPER 3 BITS IGNORED
"	10001	
"	00000	
"	01010	
"	00000	
"	10001	
"	01110	
CURSOR	00000	→ USUALLY.

You then program this pattern for the selected custom character:



```
unsigned char lcdCGData [] = { 0b01110, 0b10001, 0b00000, 0b01010, 0b00000, 0b10001, 0b01110, 0b00000 };

void main(void)
{
  // main entry point
  _DISABLE_COP();
  EnableInterrupts;

  PLL_To20MHz();
  lcd_Init();

  // enter CG Data mode
  lcd_CGChar (0, lcdCGData); // program custom character 0 to the supplied pixel pattern

  // enter DD Data mode
  lcd_AddrXY (0, 0);

  // show the custom character
  lcd_Data (0); // note, literally 0, ASCII 0-7 is reserved for custom characters

  for (;;)
  {

  }
}
```

→ SET CG ADDR THEN SEND 8 BYTES OF DATA.

You are now free to use these custom characters anywhere on the LCD.