



KodeKLIX for PUP

Advanced LCD Graphics and Animation



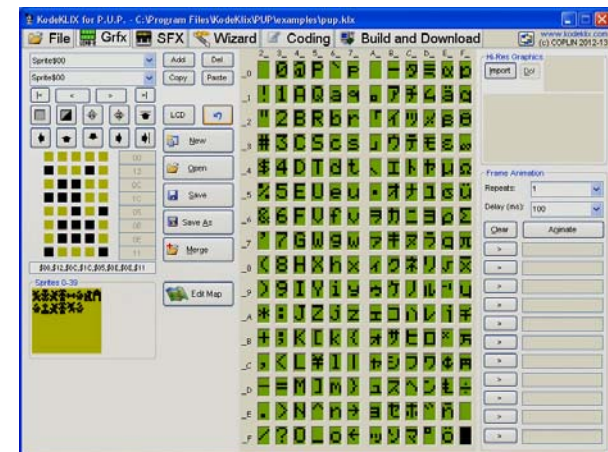
LCD Graphic and Animation

- LCD Design Module
- Animation and custom characters
- Tile based hi-resolution graphics
- Importing hi-resolution images
- PUPTM *gfx* font definitions



LCD / CGRAM Design Module

- Design new LCD character definitions
 - Equivalent to “sprites” in gaming speak
 - CGRAM = Custom Graphics RAM
- Import / convert small BMP (bitmap) images to the LCD graphics format
- Trial animation sequences
- Manage custom character libraries
 - Share these with other KodeKLIX users





LCD / CGRAM – CloseUP Guide

The screenshot shows the KodeKLIX software interface with the following callouts:

- Close**: Points to the window's close button in the top right corner.
- Sprite ID**: Points to the 'Sprite\$00' label in the top left.
- Descriptive Name**: Points to the 'Sprite\$00' text input field.
- Design Tools**: Points to the central toolbar containing icons for selection and editing.
- Design Window**: Points to the main grid area where the LCD content is being designed.
- Current Library**: Points to the 'Sprites 0-39' preview window at the bottom left.
- Built-in LCD Chars**: Points to the character set grid in the bottom right.
- Bitmap Import**: Points to the 'Hi-Res Graphics' section with 'Import' and 'Dot' buttons.
- Anim. Tester**: Points to the 'Frame Animation' section with 'Repeats' and 'Delay (ms)' settings, and 'Clear' and 'Animate' buttons.



LCD / CGRAM – Designer

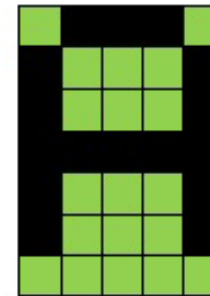
The screenshot shows the CGRAM Designer and Manager interface. Red callout boxes with lines pointing to specific UI elements are labeled as follows:

- Invert**: Points to the 'Invert' button in the top toolbar.
- Clear**: Points to the 'Clear' button in the top toolbar.
- Shift Toolbars**: Points to the 'Shift' button in the top toolbar.
- Click each Dot/Pixel**: Points to a single pixel in the grid.
- Definition Data**: Points to the hex code string at the bottom of the grid: "\$00,\$00,\$00,\$00,\$00,\$04,\$0E,\$1F".
- Library Management**: Points to the 'Add' and 'Del' buttons in the right sidebar.
- Scroll/Wrap Toolbars**: Points to the 'Copy' and 'Paste' buttons in the right sidebar.
- Undo**: Points to the 'Undo' button in the right sidebar.
- Flip Toolbar**: Points to the 'Flip' button in the right sidebar.
- File Management**: Points to the 'Open', 'Save', and 'Save As' buttons in the right sidebar.
- Line Codes**: Points to the 'Merge' button in the right sidebar.



LCD / CGRAM – Bit Codes

- Built in LCD Chars only define lines rows 1 to 7
- Custom LCD Chars consist of 8 row definitions, lowest 5bits per row define the design

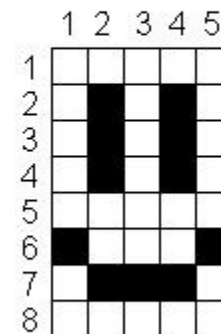


Bit: 43210 -- Hex
ROW1: 01110 -- 0x0E
ROW2: 10001 -- 0x11
ROW3: 10001 -- 0x11
ROW4: 11111 -- 0x1F
ROW5: 10001 -- 0x11
ROW6: 10001 -- 0x11
ROW7: 00000 -- 0x00

Uppercase format of Letter 'A'

A Custom 5x8 Pixel Character:

Image Coding:



Binary Coding:

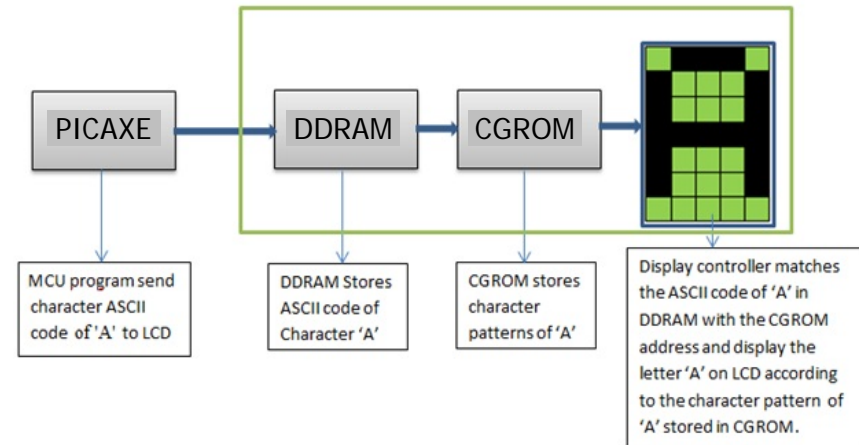
0b000 00000
→ 0b000 01010
0b000 01010
→ 0b000 01010
0b000 00000
→ 0b000 10001
0b000 01110
0b000 00000

1 = Black, 0 = White

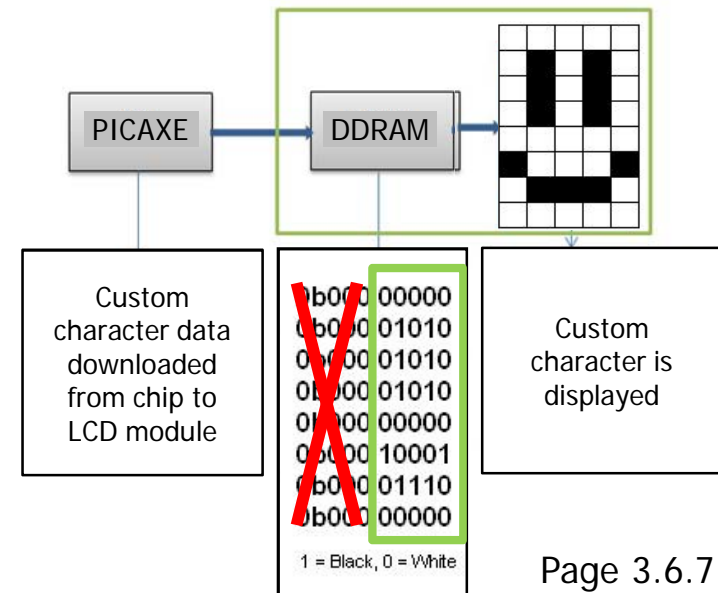


LCD / CGRAM – Bit Codes

- Built in LCD Chars only define lines rows 1 to 7



- Custom LCD Chars consist of 8 row definitions, lowest 5bits per row define the design





LCD / CGRAM – Memory

- CGRAM is memory inside the LCD control chip
- Each custom definition is stored consecutively
- Maximum space for a standard customised definitions

For 5 × 8 dot character patterns

Character Codes (DDRAM data)								CGRAM Address				Character Patterns (CGRAM data)													
7	6	5	4	3	2	1	0	5		4		3		2		1		0							
High				Low				High		Low		High				Low									
0 0 0 0 * 0 0 0								0 0 0		0 0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
												↑		↑		↑		↑		↑		↑			
												↓		↓		↓		↓		↓		↓			
0 0 0 0 * 0 0 1								0 0 1		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1		0 0 0		0 0 0		0 0 0	
												↑		↑		↑		↑		↑		↑			
												↓		↓		↓		↓		↓		↓			
0 0 0 0 * 1 1 1								1 1 1		1 0 0		1 0 1		1 1 0		1 1 1									
												↑		↑		↑		↑		↑		↑			
												↓		↓		↓		↓		↓		↓			

Character pattern (1)
 Cursor position
 Character pattern (2)
 Cursor position



LCD / CGRAM – Animation

The screenshot shows the software interface for LCD/CGRAM animation. It is divided into two main sections: a Design Window and a Frame Animation control panel.

- Design Window:** A grid of 16 frames (4x4) is shown. The top row contains 16 yellow squares. The second row contains 16 black squares. The third row contains 16 yellow squares. The bottom row contains 16 black squares. To the right of the grid is a vertical list of frame numbers: 00, 12, 0C, 1C, 05, 0E, 0E, 11. Below the grid is a sequence of hexadecimal values: \$00,\$12,\$0C,\$1C,\$05,\$0E,\$0E,\$11.
- Frame Animation Control Panel:** This panel includes:
 - Repeats: 1 (dropdown menu)
 - Delay (ms): 100 (dropdown menu)
 - Clear button
 - Animate button
 - A vertical list of 12 buttons, each with a right-pointing arrow (>).

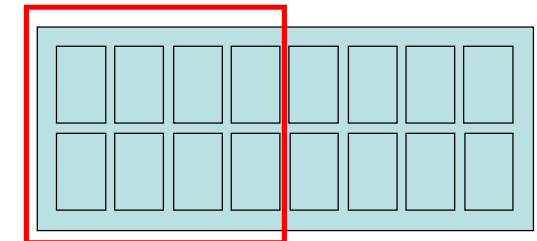
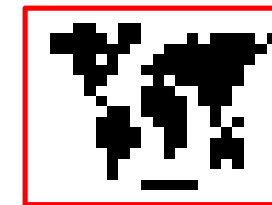
Red callout boxes with lines pointing to specific elements provide the following instructions:

- Design Window:** Points to the grid of frames.
- Playback:** Points to the frame number list.
- Clear all frames:** Points to the Clear button.
- Click to add to seq:** Points to the first arrow button in the list.
- Click for animation:** Points to the Animate button.



Tile Based Graphics – 8x2LCD

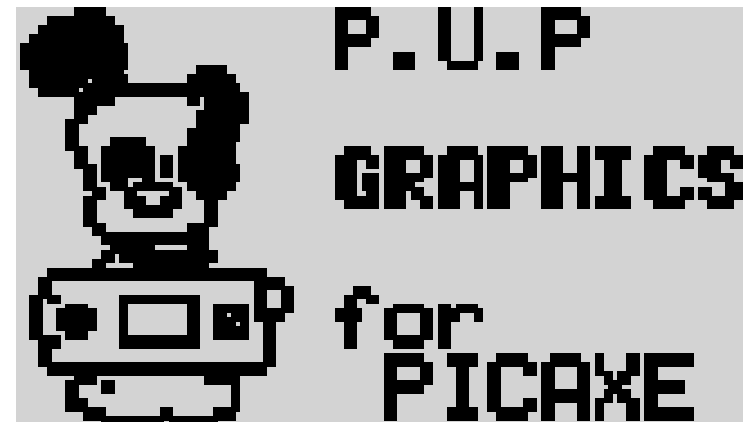
- The 8 custom characters can be drawn as “tiles” that make up a bigger picture
- The import Hi-Res graphics function lets you convert a picture into a set of tiles
- With the text LCD display there is a 1 pixel gap between display tiles





Tile Based Graphics – *gfx* LCD

- The *gfx* display can be filled with up to 84 custom characters drawn as “tiles” that make up a full screen picture
- The import Hi-Res graphics function lets you convert a picture into a set of tiles
- With the *gfx* display there is no gap between display tiles





LCD / CGRAM – Hi-Res Grfx

STEP 1

Import
20x16 BMP



Actual Size
Viewer

STEP 2

Create 8x
custom chars

Zoom &
Progress

STEP 3

Sprite
Library



New chars
added



PUP™ *gfx* – Font Def. Tables

- The *gfx* display version of PUP does not have build-in fonts
 - Fonts downloaded to PICAXE TABLE or EEPROM memory (table memory is read-only)
 - API automatically adjusts char codes for the definition table so that users can print as text
- The PUP *gfx* display can also be directly over-written combine text with graphics

```
!"#$%&'()*+,-
./0123456789:;
<=>?@ABCDEFGHI
JKLMNOPQRSTUVWXYZ
[\]^_`{|}~
```

```
abcdefghijklmnop
qrstuvwxyzç&
øßΓΠΣζηγφθΩδωφ
†‡§¨ª«¬®¯°±²³´µ¶·¸¹º»¼½¾¿
```

```
!"#$%&'()*+,-
./0123456789:;
<=>?@ABCDEFGHI
JKLMNOPQRSTUUU
XYZ[\]^_`{|}~
```

```
!"#$%&'()*+,-
./0123456789:;
<=>?@abcdefghi
jklmnopqrstuvw
xyz[\]^_`{|}~
```

```
!"#$%&'()*+,-
./0123456789:;
<=>?@ABCDEFGHI
JKLMNOPQRSTUVWXYZ
abcdefghijklmnop
```



Tutorial: 3.6
