KodeKLIX for PUP

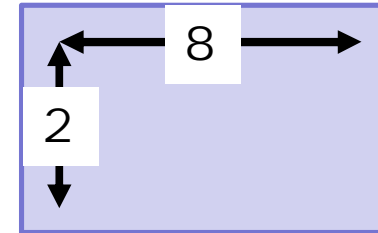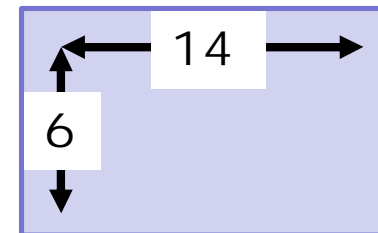## "Hello World" Example

Overview – Display Types

- **PUP original has a text module**

  - 8x2 text display

 - The 9th character printed is off the right of the screen

- **PUP *gfx* has a graphics module**

  - The KodeKLIX library simulates a 14x6 text display

  - The 15th character printed displays on the next line

"Hello World" Example App.

- Almost every BASIC tutorial begins with the "Hello World" example!

- We will do three examples in one:
  - "Hello" using a #macro command
  - "World" using the actual BASIC logic
  - Electrical "hello world" by flashing the LCD backlighting

- To begin, enable the *presets* routine…

Start: Presets ☑

Title Screen ☐
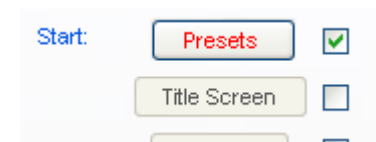
KODEKLIX

#print...#endprint Macros

- With KodeKLIX, printing is made easy with some simple #macros
    - #print1   'print top line, first position
    - #print2   'print bottom line, first position
    - #print   'print from cursor
    - #endprint   'end print macro
- 8 characters (or codes) of text printed, eg
    - ("01234567")
    - ("01234",0,1,2,3)

```
'===============================
' Initialise TITLE SCREEN
' *  Top row title
' *  Bottom row more title
'===============================
{
_DisplayTopLine:
        #print1("Hello   ")#endprint
```

Direct LCD Text Programming

- The code can also be written directly:
  - Send cursor position command
  - Then use a BASIC For...Next... loop to send each data byte sequentially
- The #print macros substitute the direct coding when the application is "built"
- Direct code is more efficient if wanting to send fewer bytes, or if you want format control

```
_DisplayBottomLine:
        _byte1=$C0
        gosub _SendCmdByte
        for b0=0 to 7
          lookup b0,("World   "),_byte1
          gosub _SendDataByte
        next
```

LCD BackLight Programming

- The LED BackLight for the LED screen is controlled through PICAXE port pin a.0

- Pin a.0 can be set HIGH, LOW or TOGGLE
  - When LOW, the LED is switched "on" (this is called low-side driven)

- The DO...LOOP... shows how efficiently a never ending loop can be coded

- WAIT 1 introduces a pause between TOGGLE events.

```
do ' Loop forever
        toggle a.0
        wait 1
        loop
```

Tutorial: 3.1

- Open tutorial 3.1

- Study code snippets for:
  - Presets
  - Title Screen

- Connect PUP and Download program

- What do you observe on the screen?

  - _____

  - _____