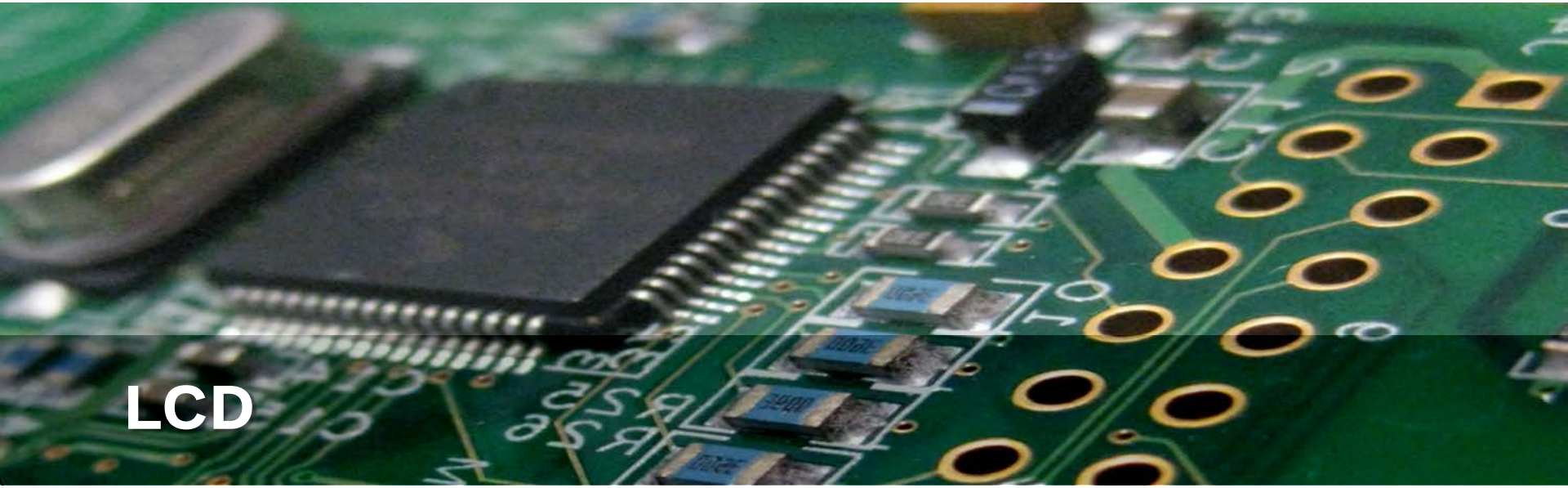
Lucerne University of
Applied Sciences and Arts
HOCHSCHULE
LUZERN
Technik & Architektur



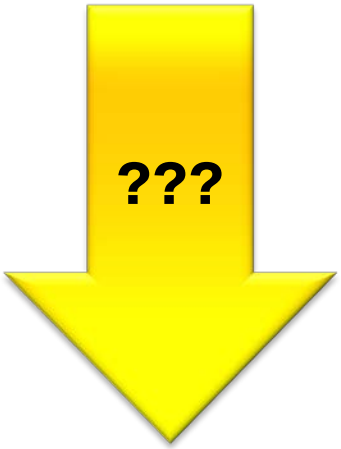
LCD

“Mirror, mirror, on the wall, ...”

Prof. Erich Styger
erich.styger@hslu.ch
+41 41 349 33 01

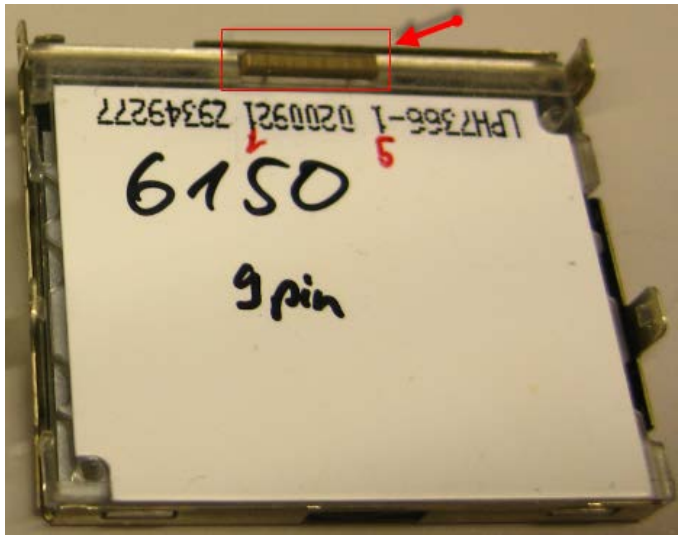
Learning Goals

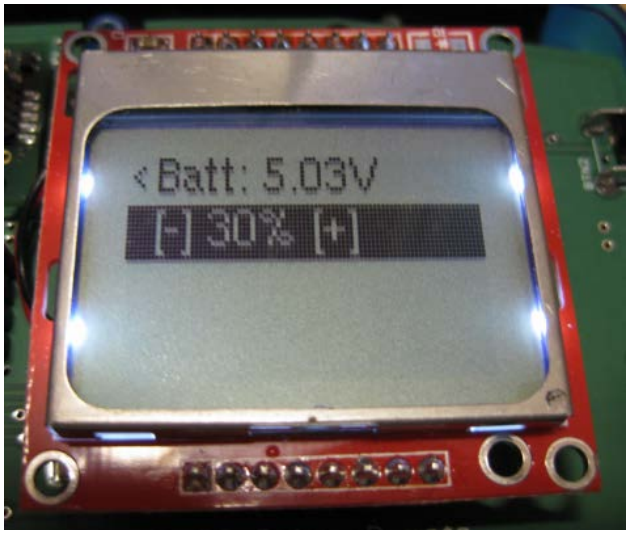
- LCD Display memory
- Nokia LCD Display
- Using the LCD on the remote board



Nokia Displays

- Simple, large volume, low cost
- SPI interface (Philips PDC8544 display controller)
- 2nd Life Recycling
  - ‘Zebra’ connector
  - Mounted on PCB with backlight LEDs
- https://mcuoneclipse.com/2012/12/16/zero-cost-84x48-graphical-lcd-for-the-freedom-board/

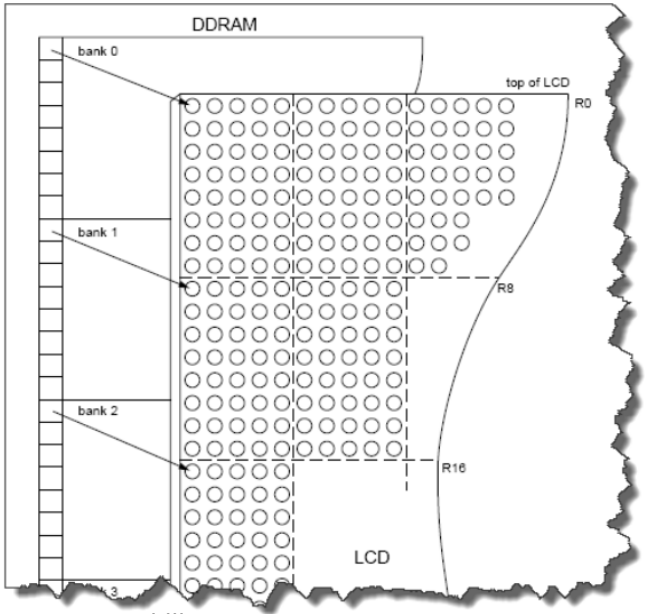




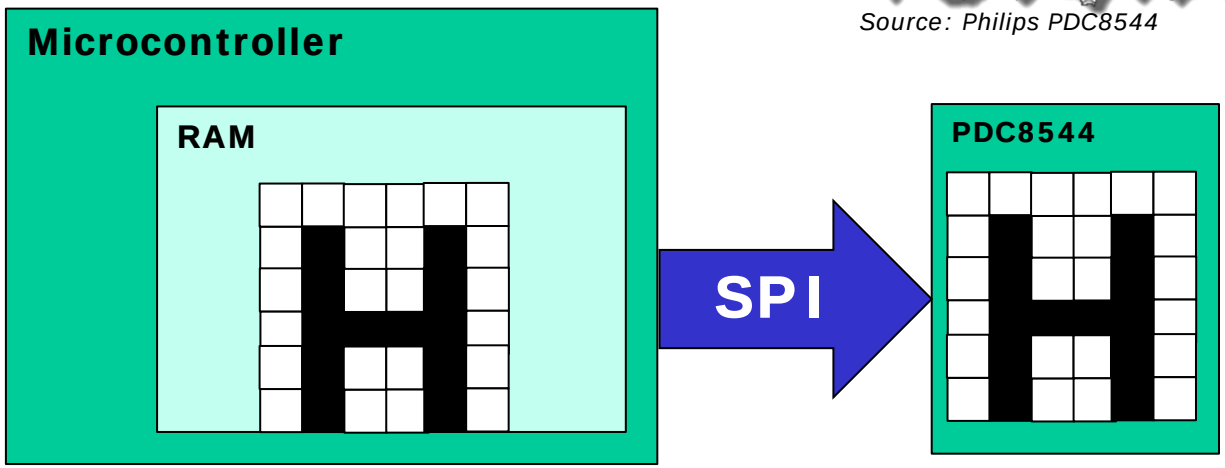
PDC8544

DISPLAY: 64 x 48 Pixel

- SPI interface (only MOSI)
- Cannot read display memory
- Draw operation in microcontroller RAM
- Write/Update display controller RAM
- Defined Sequence, Endianess, Orientation of bits in memory

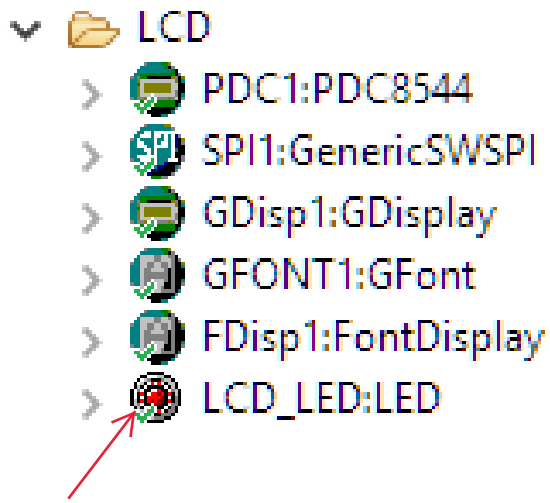


Source: Philips PDC8544



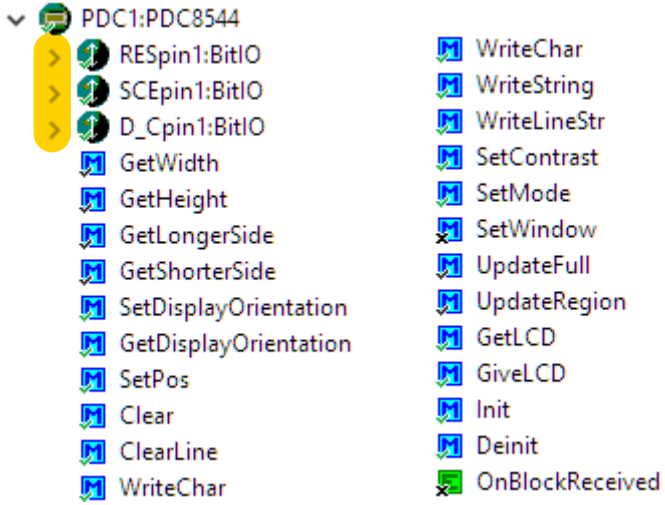
LCD Components

- PDC8544
  - Philips PDC8544 display driver
- SPI
  - Display uses SPI
  - bit banging because of available pins
- GDisplay
  - graphical display routines (line, circle, ...)
- GFont
  - graphical fonts
- FontDisplay
  - Font writing routines
- LCD_LED
  - Background illumination



PDC8544

- Low Level Display Driver
- Memory mapping (bit/byte order)
- Display properties (size) and critical section handling
- Includes basic character string writing (line based)
- `PDC1_WriteLineStr(1, "hello");`



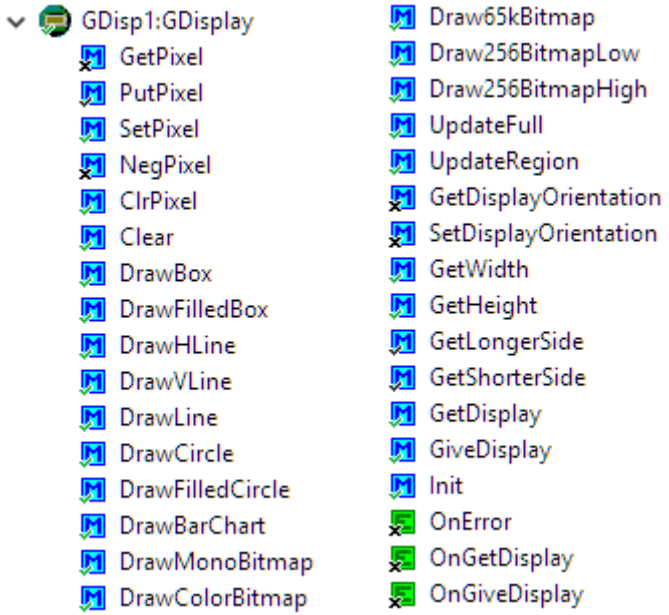
Don't change them!

Name	Value
Component name	PDC1
Properties	
Width	84
Height	48
Bytes in rows	no
Bytes in x direction	yes
MSB first	no
Bits per pixel	1
Window capability	no
Display Memory Write	no
Type	LPH7366
Invert Display	no
Initialize contrast	Disabled
Mode	normal
Voltage	V3
HW non-LDD	Enabled
RES	RES
SCE	SCE
D_C	D_C
SPI	Enabled
SPI	SPI1
HW LDD	Disabled
System	
Wait	WAIT1
Initialize on Init	yes

GDisplay

no reading from display!

- Display drawing routines
- Set/Clear/Put/Get Pixel
- Display orientation
- Implements memory buffer



Name	Value
Component name	GDisp1
Inverted Pixels	no
Memory Buffer	Enabled
Orientation	Landscape
Clear screen on Init	no
Hardware	
Display	PDC1
Watchdog	Disabled
RTOS	Disabled

possible to change at runtime

GDisplay Example

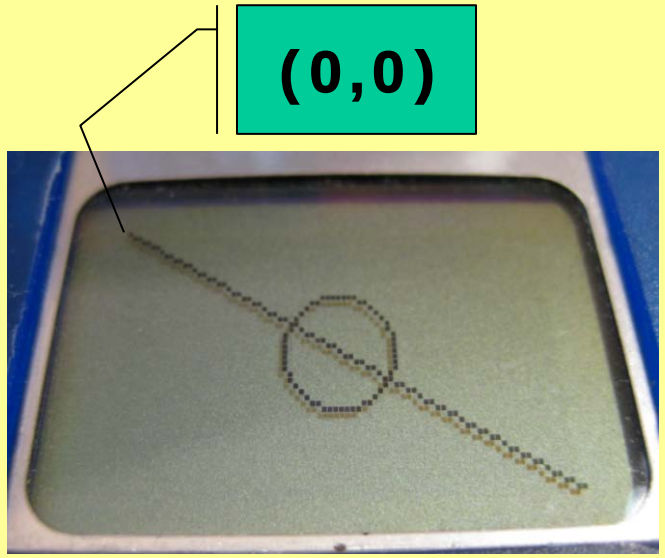
- Drawing into microcontroller memory
- UpdateFull(): write memory to display

it's also possible to update just one region (UpdateRegion) -> it's a bit faster!!

```
GDisp1_Clear();
GDisp1_UpdateFull();

GDisp1_DrawLine(0, 0,
  GDisp1_GetWidth(), GDisp1_GetHeight(),
  GDisp1_COLOR_BLACK);

GDisp1_DrawCircle(
  GDisp1_GetWidth()/2,
  GDisp1_GetHeight()/2,
  10, GDisp1_COLOR_BLACK);

GDisp1_UpdateFull();
```

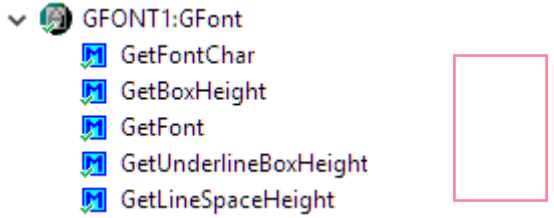
clear the display buffer

write the display-buffer

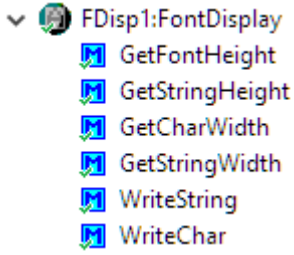


FontDisplay & GFont

- Driver to write with fonts
  - WriteChar()
  - WriteString()
- Uses ‘font handle’: Pointer to font descriptor
  - GetFont()
- One component instance for font/size
- Variable character size



Name	Value
Component name	GFONT1
Overwrite Bounding Box Heigh	Disabled
Name	Helv
Size	8
Style	normal



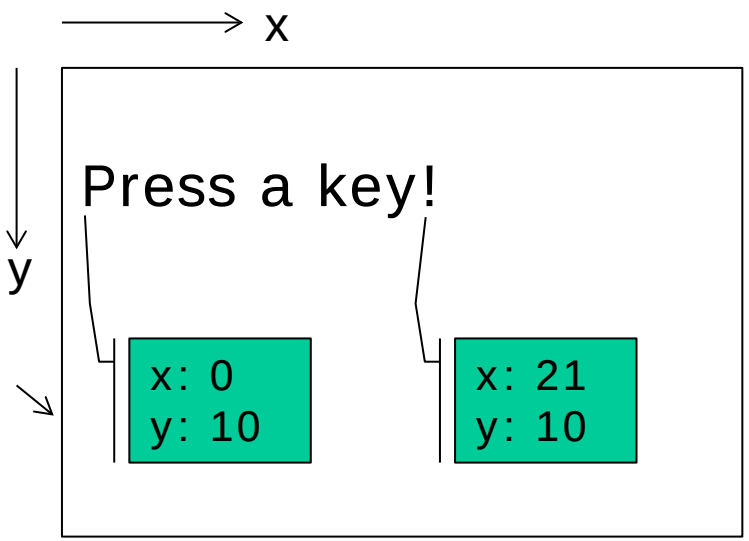
Name	Value
Component name	FDisp1
System	
Display	GDisp1
Font	GFONT1
Watchdog	Disabled

Font & GFont Example

- x and y position passed by reference
- Incremented/changed for successive writing

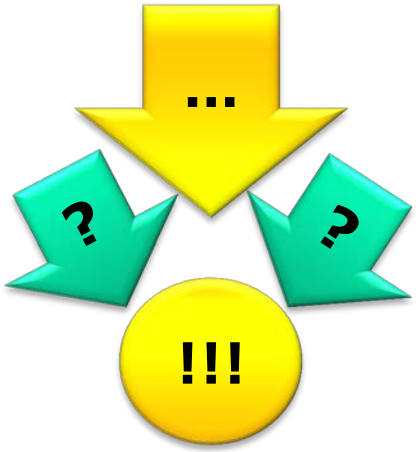
```
FDisp1_PixelDim x, y;

x = 0;
y = 10;
FDisp1_WriteString(
    "Press a key!",
    GDisp1_COLOR_BLACK,
    &x, &y,
    GFONT1_GetFont());
GDisp1_UpdateFull();
```

pointer to where starting. why a reference?



Summary

- LCD Memory map
- Low Level Display Driver
- Graphic Drawing Primitives
- Text and Fonts



Lab: Nokia Display

- Use the Display
  - Draw Text
  - Draw Graphics
  - Write with Fonts
  - Display Button Status