Touch and graphical displays part 2 - Display programming

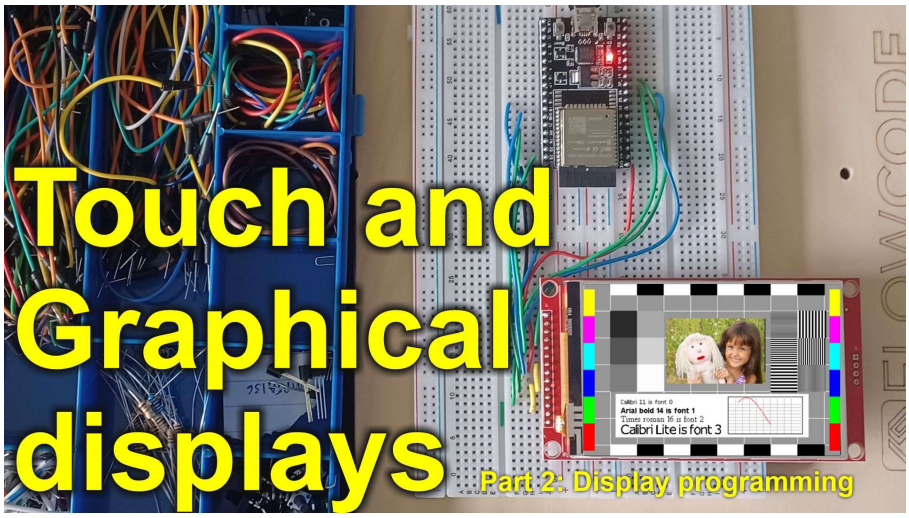


Figure 1 - Youtube video link

1. The project
2. The Hardware
3. The Software
4. Conclusion
5. Resources

The project

In this article we are going to look at how to program touch screen displays using the free version of Flowcode. In this case we are going to make a test card which includes various coloured shapes, includes a bitmap image, use different fonts of texts and draws a graph. This test card is not a project goal in itself: but you can use the techniques here in your other projects and a test card like this allows us to set a benchmark for display performance. We are going to look at two types of graphical display interface: a SPI serial bus and a parallel bus and we will compare the performance of the two.

The software we are using is Flowcode: the simulation features of Flowcode makes it very easy to design touch screen systems :

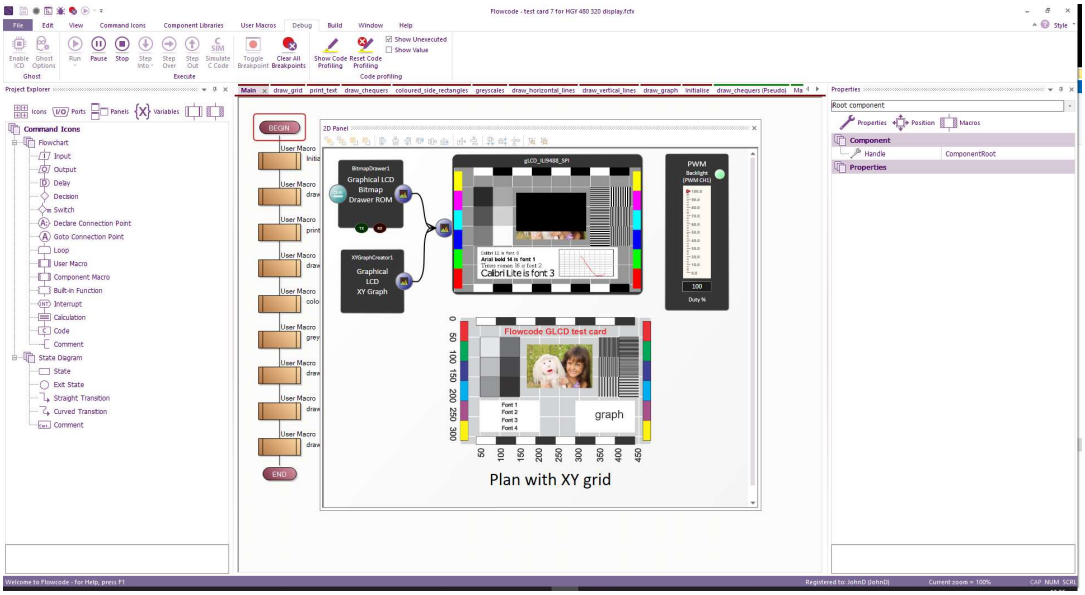


Figure 2 - final Flowcode program

The hardware

Copyright © 2024 Matrix TSL

We will be using an ESP32 wroom development board with a separate SPI bus display on a prototype board. This uses a 480 x 320 resistive touch screen colour graphical display. You don't have to use this processors or display: Flowcode supports thousands of microcontrollers and many different displays including ST ARM, many types of PIC, Arduinos and more.

Figure 3 - touch screen front

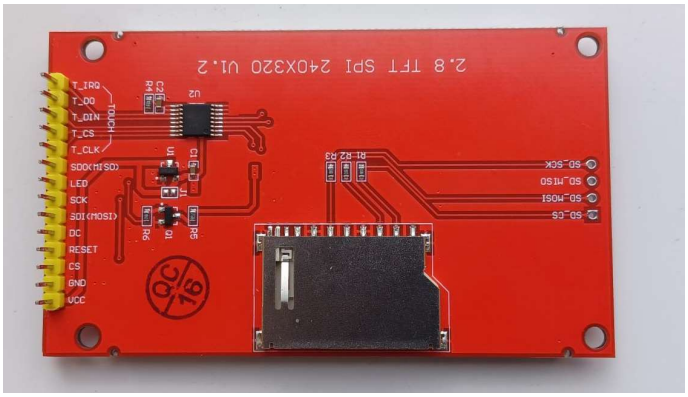


Figure 4 - touch screen rear

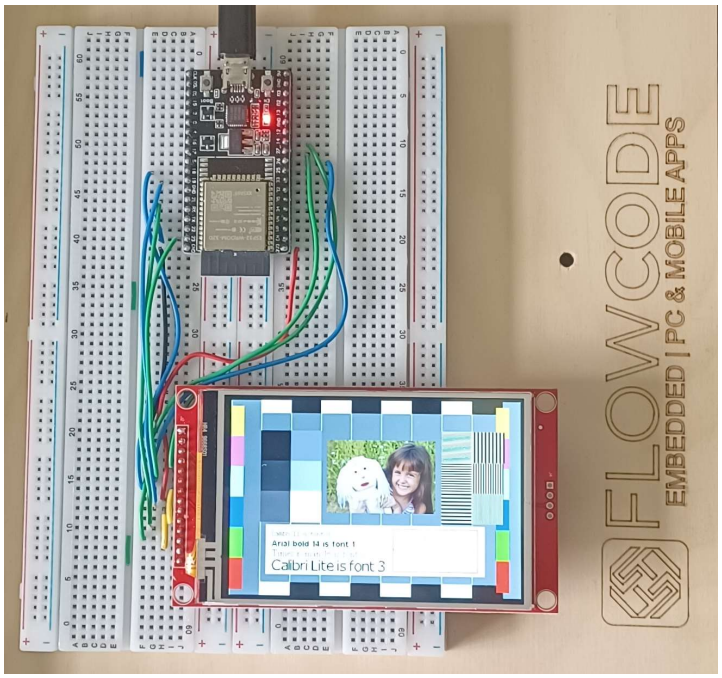


Figure 5 - breadboard with ESP32 Wroom and touch screen

For this project we are using a touch screen bought from Amazon and described as : "HGY LCD Module TFT Touch Screen Display Serial Peripheral Interface ILI9488 HD 480x320 3.5in". You can see the connections to it on the rear of the PCB. There are two groups of connections: ILI9488 graphical display connections and XPT2046 touch screen chip. You can see the final project in figure 5.

Copyright © 2024 Matrix TSL

The connections between the touch screen and the ESP32 Wroom are as follows:

Touchscreen		ESP32 Wroom32D	
Pin 1	VCC	3V3	
Pin 2	GND	GND	
Pin 3	CS (LCD)	IO27	PORTA.27
Pin 4	RESET	IO26	PORTA.26
Pin 5	DC (LCD)	IO14	PORTA.14
Pin 6	MOSI (LCD)	IO23	PORTA.23
Pin 7	SCK (LCD)	IO18	PORTA.18
Pin 8	LED background 3V3 (PWM)		
Pin 9	MISO (LCD)	IO19	PORTA.19
Pin 10	SCK (touch)	IO18	PORTA.18
Pin 11	CS (touch)	IO21	PORTA.21
Pin 12	MOSI (touch)	IO23	PORTA.23
Pin 13	MISO (touch)	IO19	PORTA.19
Pin 14	Not connected		

So we have a SPI bus connected to both the Touch screen chip and the display driver chip with separate CS enable lines.

The software

Before developing any graphical display system you need a plan. It is possible to develop the system through continuous cycles of design and coding, but this will take a lot of time. Having a clear plan of what you want to achieve before starting will help you a lot.

For this project we designed our touch screen layout in Corel. It is important that you include an XY grid in your plan to give you a good idea of the location of all of the elements of the design. You can export your design into a BMP file (not JPEG) so that it can be loaded on the Flowcode panel as a design guide. We are going to design the basic software, simulate it to check its functionality, and then we will deploy it on the two hardware platforms.

Here is the plan:

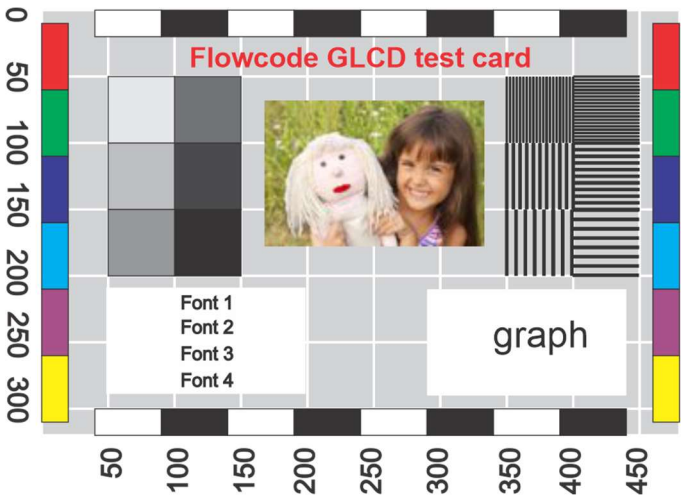


Figure 6 - touch screen plan.

Then start up Flowcode Embedded. The processor type is ESP 32_WROOM_32. Add a new 2D panel and then select COMPONENT LIBRARY... CREATION...IMAGE. In the Image properties add the plan graphic that you made earlier.

Copyright © 2024 Matrix TSL

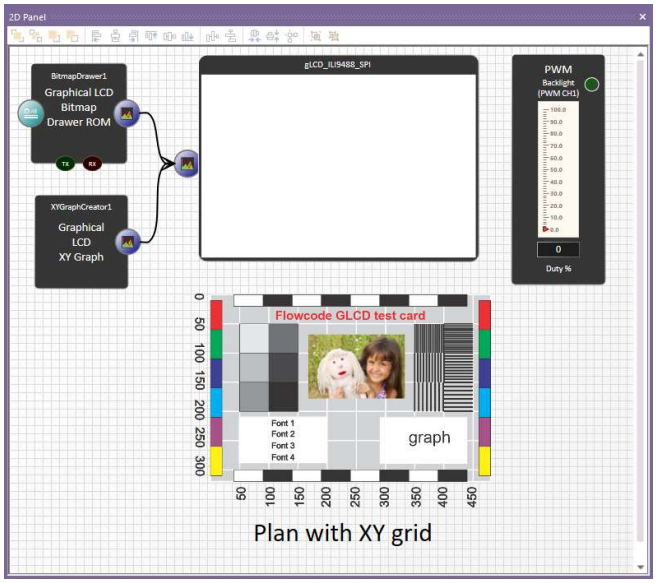


Figure 7 - Flowcode panel design.

Then add a ILI9488 display, Graphical LCD bitmap drawer, Graphical LCD XY graph and PWM component for screen brightness.

Link the Graphical LCD bitmap drawer and Graphical LCD XY graph to the ILI9488 display in the properties of each component. Next you need to make the virtual connections to the ESP32 microcontroller. First let's look at the display:

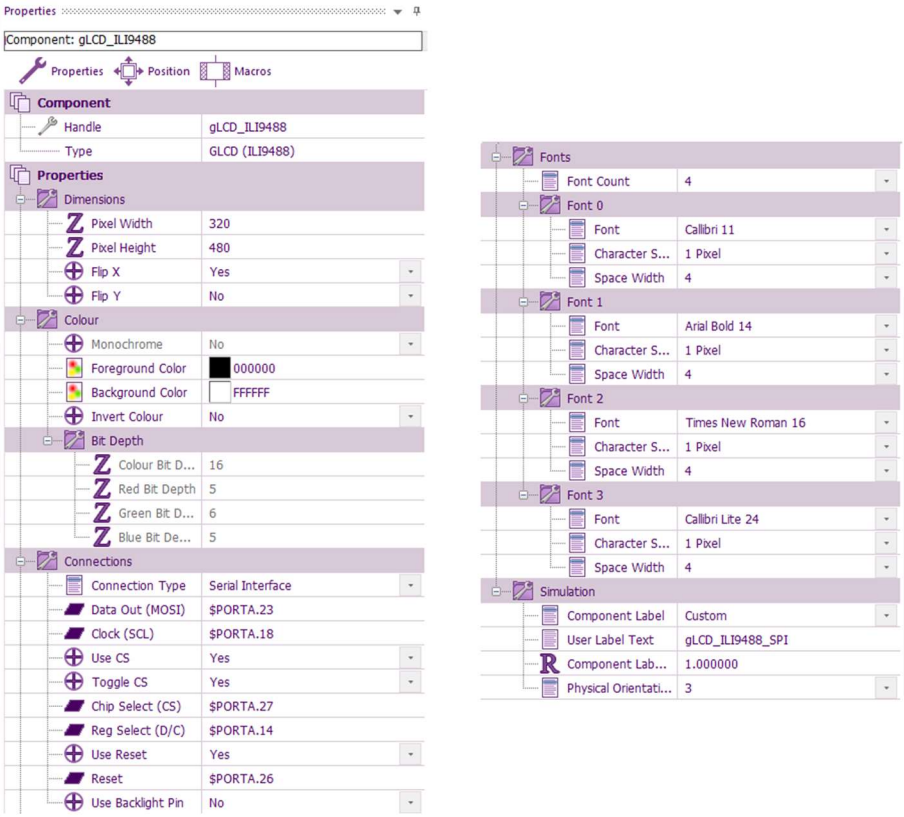


Figure 8 - ILI9488 display Properties

Copyright © 2024 Matrix TSL

Here you can see the ILI9488 properties. A few things of note: You can select different orientations and X Y pixel counts - the ILI9488 is capable of multiple resolutions. The ILI9488 is an SPI bus controlled device. You can set up how this connects to the ESP32 pins in the Connections section of the Properties. You can embed up to 4 fonts in the display program. You can choose the fonts in the system or have custom fonts if you provide a file. The properties set up the fundamentals of the touch display and set the connections to it.

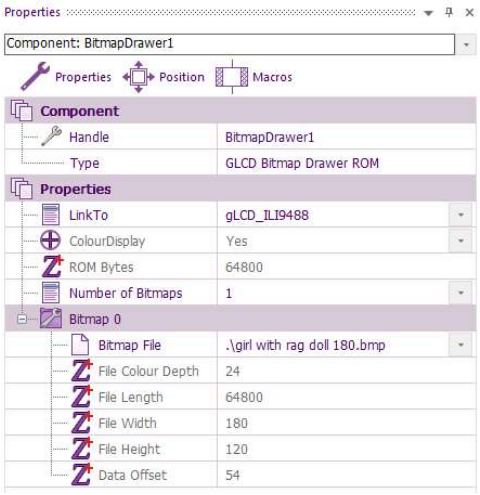


Figure 9 - Bitmap drawer properties

In Fig 9 you can see the Bitmap drawer properties. This is quite simple: you select the file you want and this component makes sure that the bitmap is sent into the ESP32 chip and stored in ROM ready for the program to load and send to the graphical display. Finally you set up the graph:

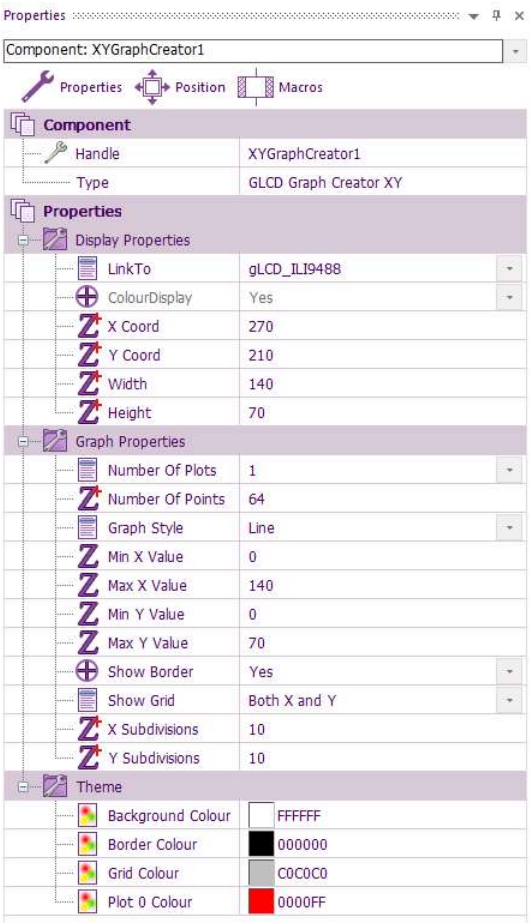


Figure 10 - XY graph properties

Copyright © 2024 Matrix TSL

Now you are ready to start to write your program.

Developing text and graphics for the display is carried out through hardware macros. Highlight the ILI9488 display component on the panel and select COMPONENTS in the Project Explorer on the left of the screen. Expanding the ILI9488 entry you can see all the hardware macros you have available to you as shown in Figure 11:

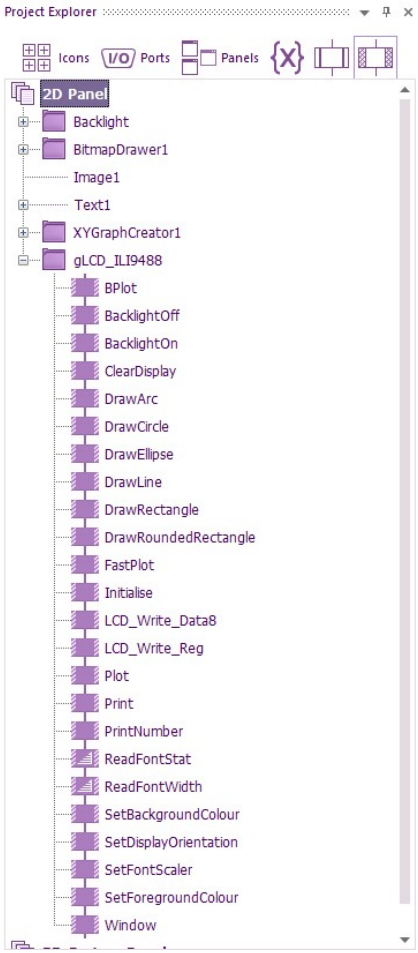


Figure 11 - hardware macros of the ILI9488 display

You need to initialise the display - so drag an Initialise component into your program. Then you select the Foregroundcolour and add it to your program with Red=0, Green=0, Blue = 0 :

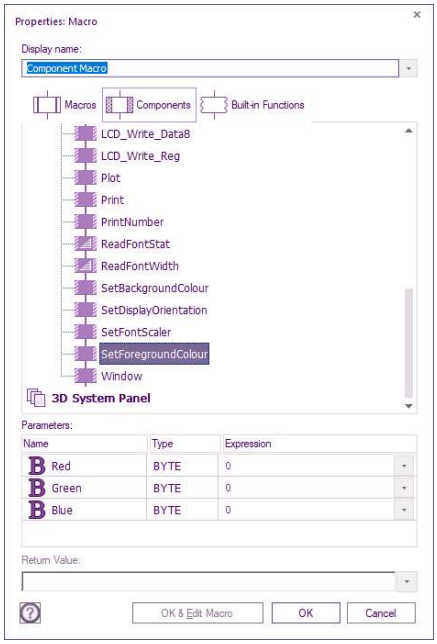


Figure 12 - Macro Foregroundcolour dialogue box

Copyright © 2024 Matrix TSL

Then add DrawRectangle onto your program to give you a dialogue box: as follows:

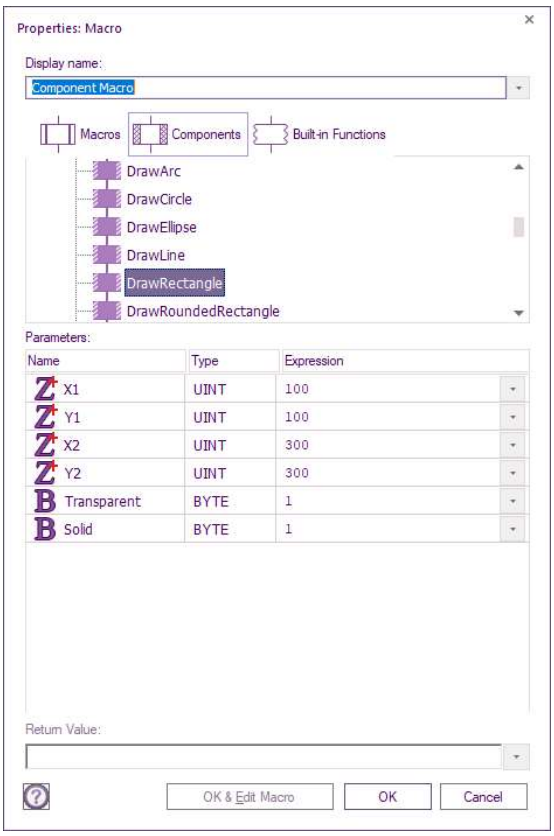


Figure 13 - Macro DrawRectangle dialogue box

Then select DEBUG….RUN. This is the result:

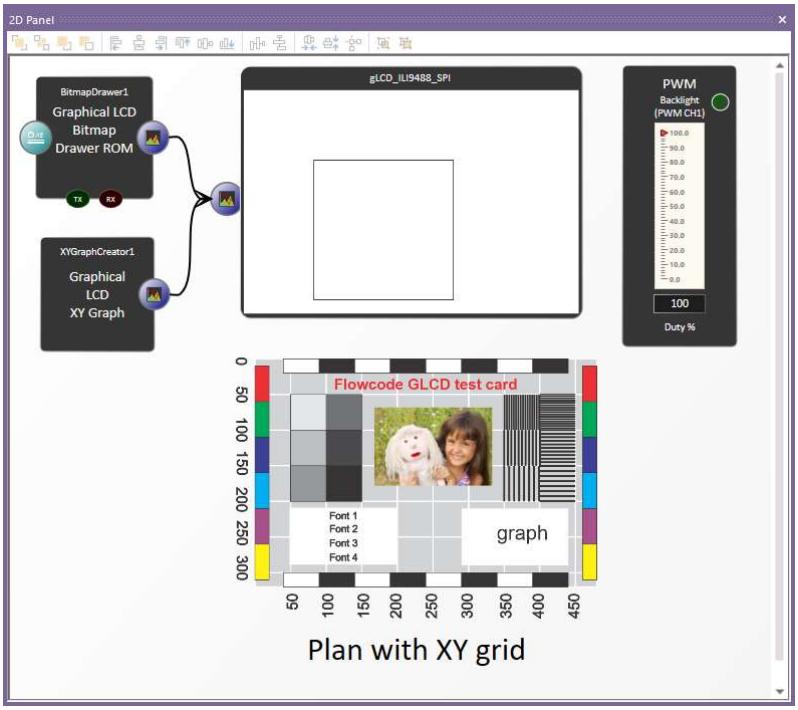


Figure 14 - your first shape

Then you need to compile this to the hardware. First select BUILD...PROJECT OPTIONS and make sure the PROGRAMMER PORT is set up correctly. If an option does not come up you may need to load a driver from the Espressif site. Then select BUILD...COMPILE TO TARGET. If you have everything set up correctly then you will get a rectangle on your display.

Now that you understand how the macros are used you should be able to design the test card which just consists of a sequence

Copyright © 2024 Matrix TSL

of shapes, lines and bits of text. Here is the code for drawing the black and white chequers on the top and bottom of the display - shown in Pseudocode this time rather than flowcharts:

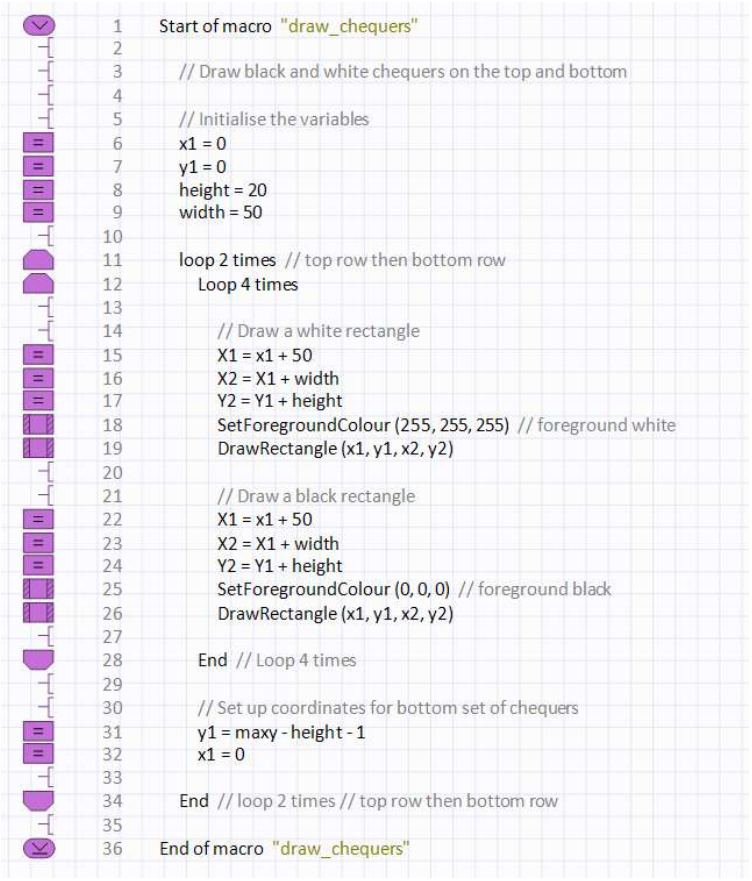


Figure 15 - Draw chequers macro in Pseudocode

And here is the main program:

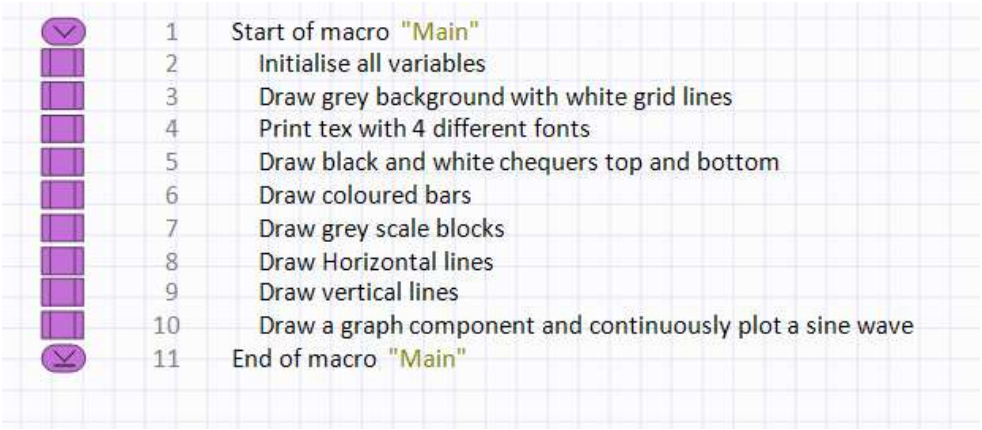


Figure 16 - Main program in Pseudocode

The Graph macro sets up a graph and continuously plots a sinewave with limited number of points.

We will not go through all of the macros here: you can either have a go at recreating them using the information above, or you can download the completed program from the link given in the Resources section at the bottom of the document. The information above should give you a good steer on how to design your program if you fancy the challenge.

Flowcode simulates the whole program. The following image shows the simulated image:

Copyright © 2024 Matrix TSL

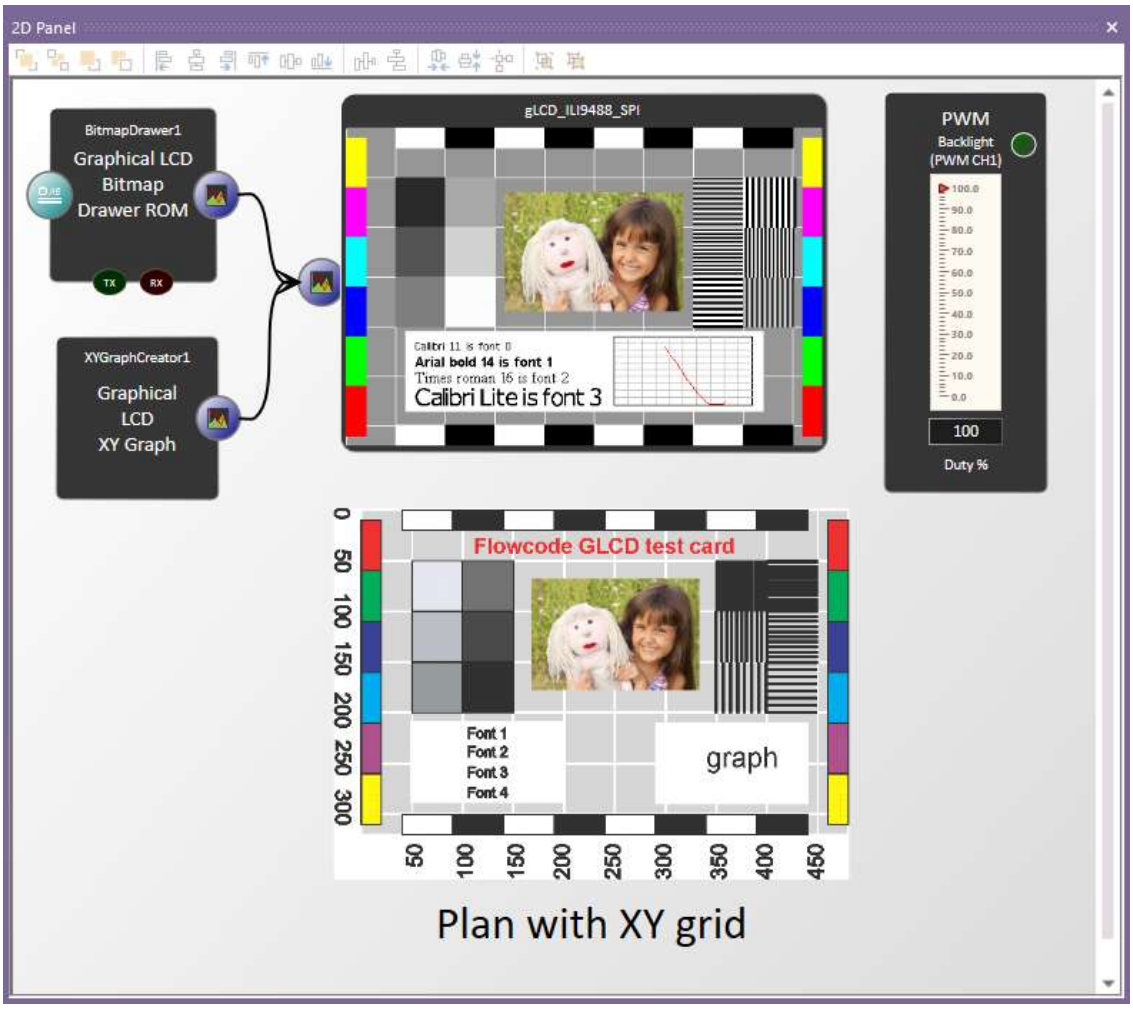


Figure 17 - Final simulated display

If you are building the program from scratch then it would be a good idea to repeatedly build the program and send it to the hardware as you go.

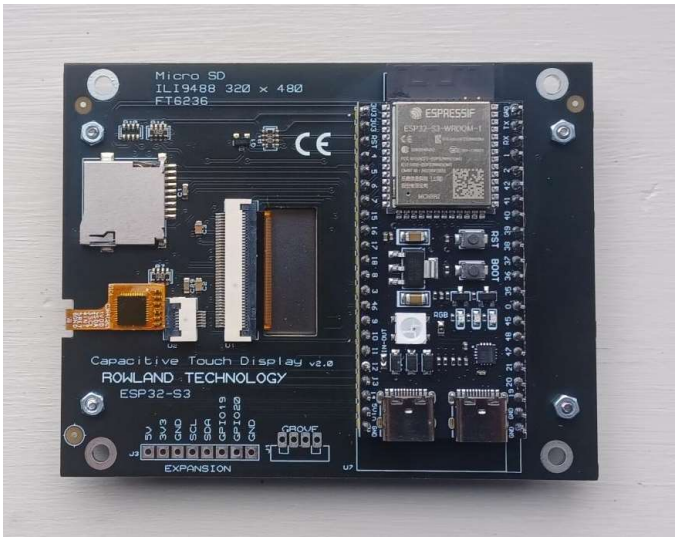
This works very well. But the time taken to draw the display is around 20 seconds. This is the problem with SPI addressed Graphical displays: the SPI bus is quite slow.

Lets take a look at an alternative. The Rowland Technology display has an ESP32 WROOM32 and a 480 x 320 graphical touch screen display built int to it.

Figure 18 - Rowland technology GLCD - front

Copyright © 2024 Matrix TSL

Figure 19 - Rowland technology GLCD - rear



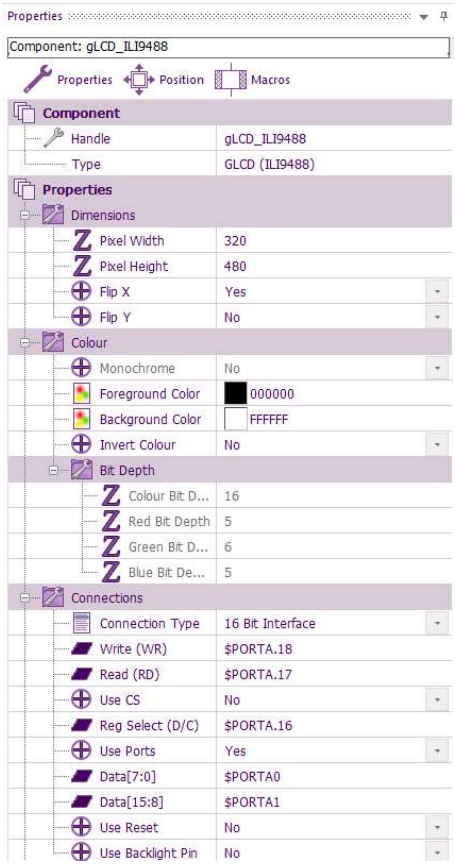
From the photo of the rear of the board you can see the ribbon cables of the graphical display and the FT6236 touch chip. The display is connected using 16 parallel wires. That allows data to be sent to tit much faster. The ESP32-S3 chip is has a similar performance to the ESP32-WROOM32 used on the prototype board. The connections are as follows:

Touchscreen	ESP32 Wroom32D
Data 0-15 (LCD)	IO0-IO15
Reg Select or D/C (LCD)	IO16
Read (LCD)	IO17
Write (LCD)	IO18
Backlight (LCD)	IO21
Chip Select (SD card)	IO38
MOSI (SD card)	IO45
MISO (SD card)	IO35
SCK (SD card)	IO46
Card Detect (SD card)	I42
Interrupt (Touch)	IO39
SCL (Touch)	IO47
SDA (Touch)	IO48

So you can see that the connection to the graphical display is a 16 bit parallel connection, the touch screen chip is I2C and the SD card is SPI.

Figure 20 - ILI9488 properties for parallel connection

Copyright © 2024 Matrix TSL



In the resources section you will see that we have written two programs: one for the SPI bus display with ESP32 WROOM32 on the prototype board, and one for the Rowland Technology display with the ESP32-S3 and the parallel connected ILI9488 display.

The key difference is the performance of the final hardware: the parallel connected display takes just over 1 second to draw the display versus 20 seconds for the SPI display.

Conclusions

- The simulation facilities of Flowcode make it a great tool for designing graphical displays.
- Using the SPI bus to communicate display data is very easy from a connection and prototyping point of view.
- Using the SPIbus is very slow and is likely only to be useful in simple displays without graphics.
- Using the parallel bus for graphical displays is very fast .
- To take advantage of the parallel bus you need to either make your own PCB with Flexi circuit board connections of you need to use an off-the-shelf module from Rowland Technologies, Elecrow or other manufacturer.
- If you use an off the shelf graphical display module then moving to a two processor solution for your electronic sytsems may be the way forward.

Resources

The programs developed in this article can be downloaded from the Flowcode Wiki:

https://www.flowcode.co.uk/wiki/index.php?title=Graphical Colour Displays

Copyright © 2024 Matrix TSL