ESP32 draadloze spelletjes (ESP-Mesh)

Program on github:

https://github.com/thieu-b55/ESP32-Wireless-Touch-Screen-Games

Play:

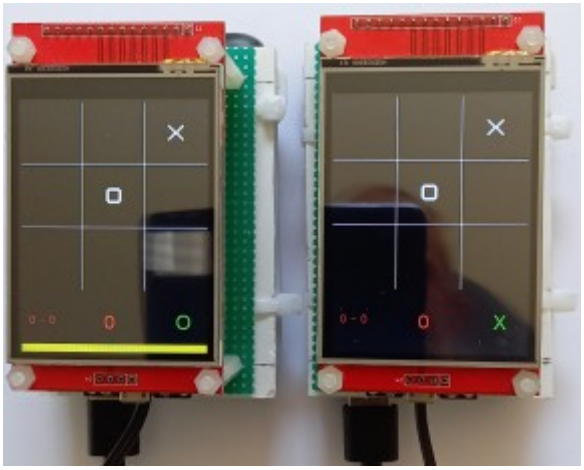
TIC-TAC-TOE

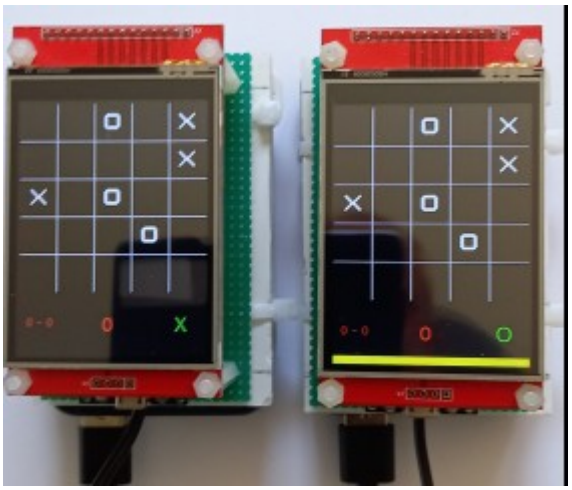
Four In A Row

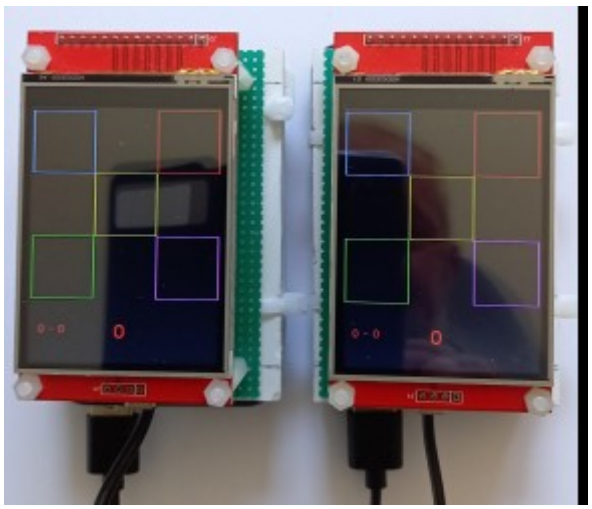
Memory game

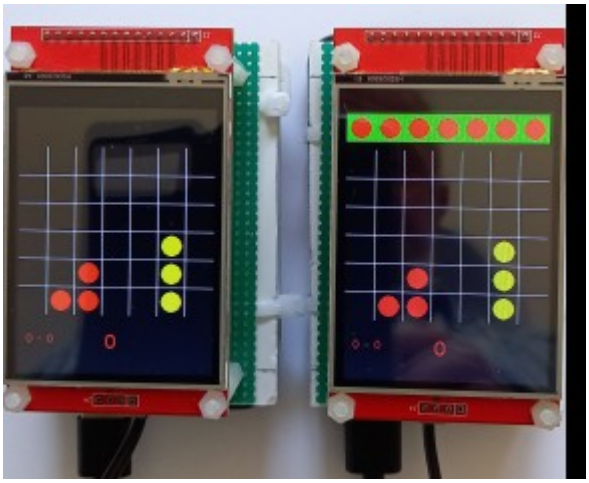
Four Coins Falling In A Row

With 2, connected to each other via a Mesh network, ESP32 modules









Components

WROOM Devkit module
2x 2.8 TFT SPI 240x320 TFT **TOUCH** SCREEN

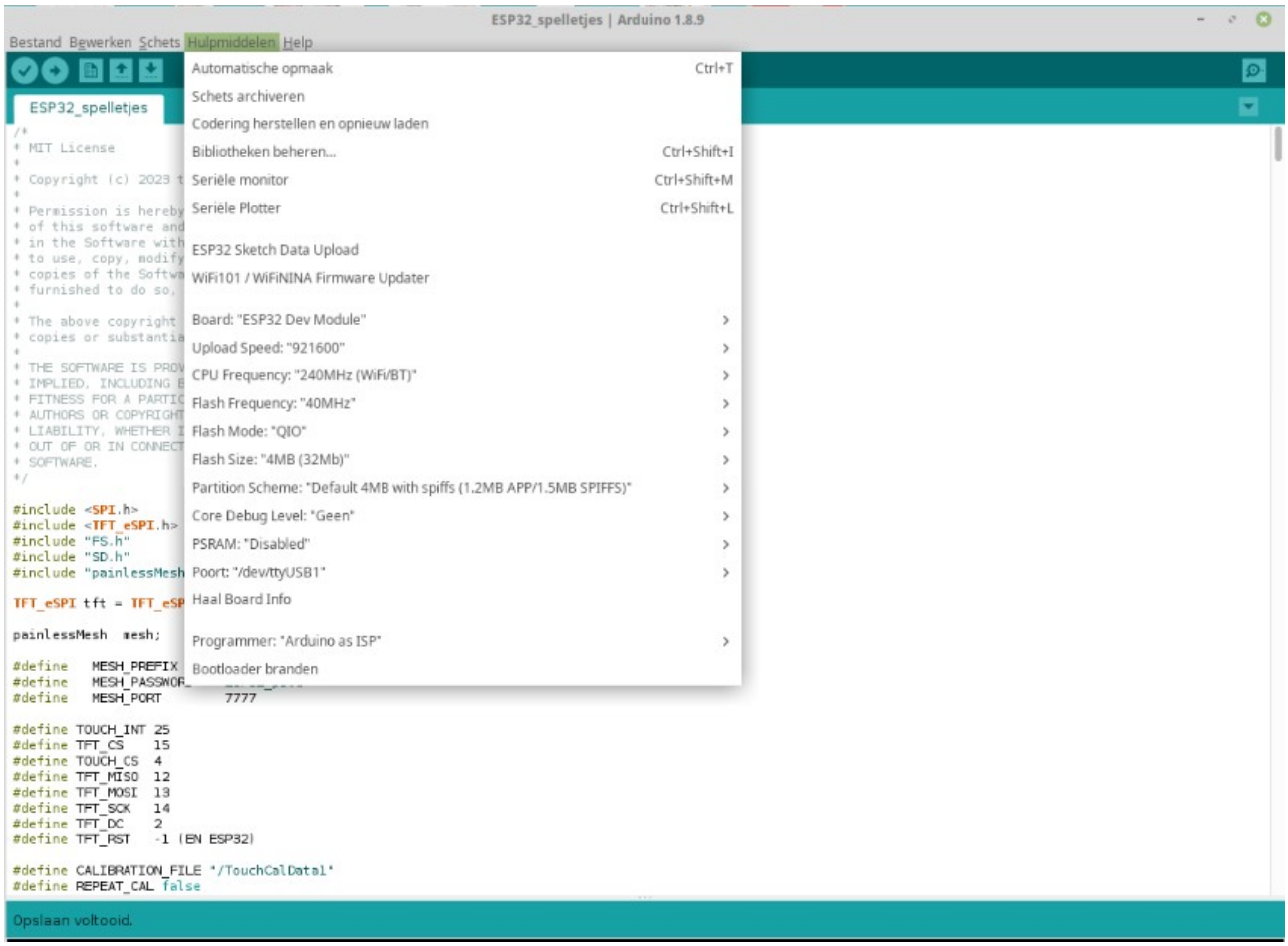
Connections

ESP32	5V	>>	5V voeding
ESP32	GND(3x)	>>	GND voeding
ESP32	EN(Reset)	>>	TFT RESET
ESP32	2	>>	TFT DC
ESP32	4	>>	T_CS
ESP32	12	>>	TFT SDO(MISO)
			T_DO
ESP32	13	>>	TFT SDI(MOSI)
			T_DIN
ESP32	14	>>	TFT_SCK
			T_CLK
ESP32	15	>>	TFT CS
ESP32	25	>>	T_IRQ
ESP32	3.3V	>>	TFT LED

Settings User_Setup.h TFT_eSPI

```
// ###### EDIT THE PIN NUMBERS IN THE LINES FOLLOWING TO SUIT YOUR ESP32 SETUP   ######

// For ESP32 Dev board (only tested with ILI9341 display)
// The hardware SPI can be mapped to any pins

#define TFT_MISO 12
#define TFT_MOSI 13
#define TFT_SCLK 14
#define TFT_CS   15   // Chip select control pin
#define TFT_DC    2  // Data Command control pin
#define TFT_RST  -1  // Reset pin (could connect to RST pin)
//#define TFT_RST  -1  // Set TFT_RST to -1 if display RESET is connected to ESP32 board RST

// For ESP32 Dev board (only tested with GC9A01 display)
// The hardware SPI can be mapped to any pins

//#define TFT_MOSI 15 // In some display driver board, it might be written as "SDA" and so on.
//#define TFT_SCLK 14
//#define TFT_CS   4  // Chip select control pin
//#define TFT_DC   2  // Data Command control pin
//#define TFT_RST  -1  // Reset pin (could connect to Arduino RESET pin)
//#define TFT_BL   22  // LED back-light

#define TOUCH_CS 4     // Chip select pin (T_CS) of touch screen
```

Settings Arduino IDE



Load the program ESP32_spelletjes.ino into both ESP32 modules.

If necessary, adjust the MESH_PORT if you want to use more than 1 set simultaneously.

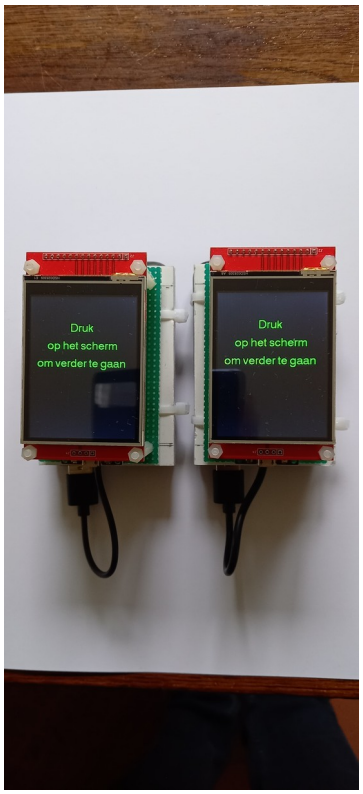
```
painlessMesh  mesh;

#define   MESH_PREFIX     "ESP32"
#define   MESH_PASSWORD   "ESP32_pswd"
#define   MESH_PORT        7777
```

2 matching modules must have the same MESH_…. data.

How does it work

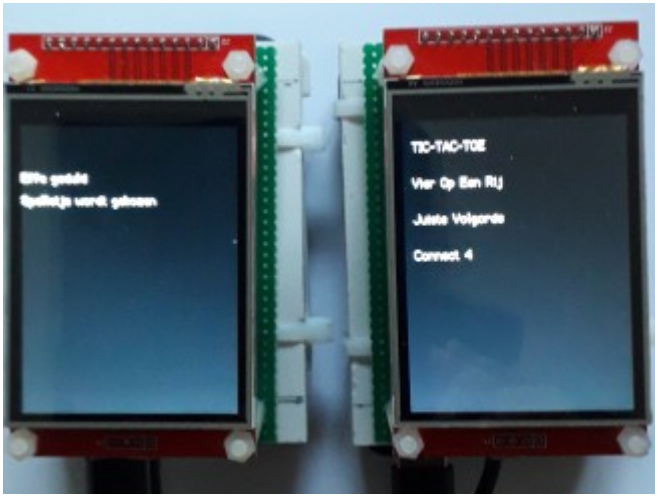
This screen appears after startup

Both ESP32 modules now choose a random number and wait for connection with the opponent. After connection, the following screen will appear



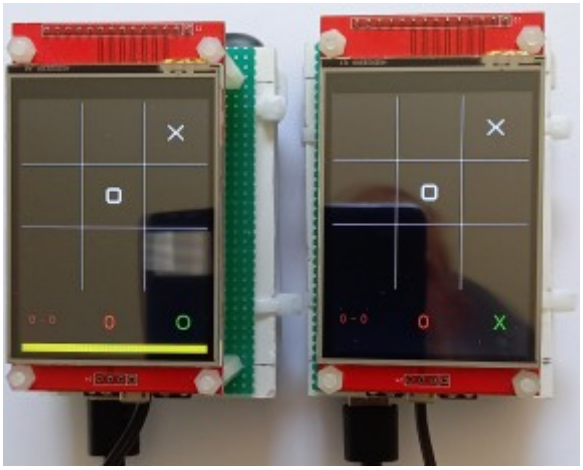
1 of the two players must now touch the screen (the supplied touch pen gives the best result). Who has no influence on the further course, this is determined by the highest randomly chosen number of the ESP32 modules

After touching the screen, the following screens appear



The module with the highest random number may choose which game to play and may also start first. After each game, it is switched who may start.

TIC-TAC-TOE

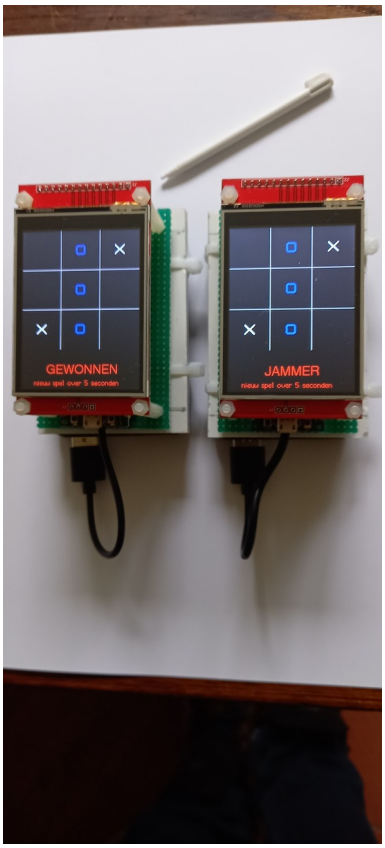


View at the bottom of the screen
Left
scoring with own points first and then the opponent

Middle
own points again

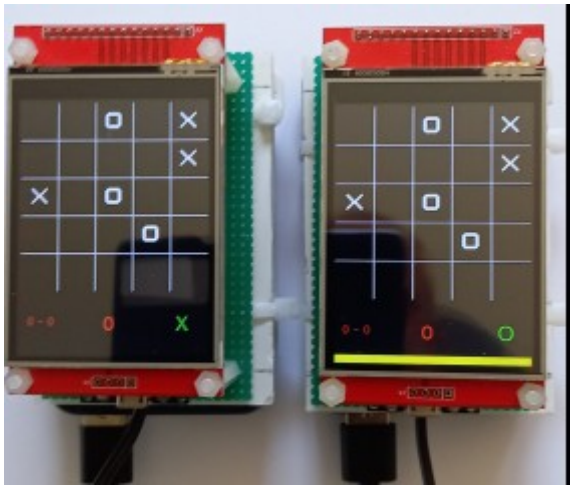
Right in green
which letter you are playing with

Yellow bar indicates whose move it is.



A correct combination is shown in blue.
A new game starts after 5 seconds.

Four In A Row



View at the bottom of the screen
Left
scoring with own points first and then the opponent

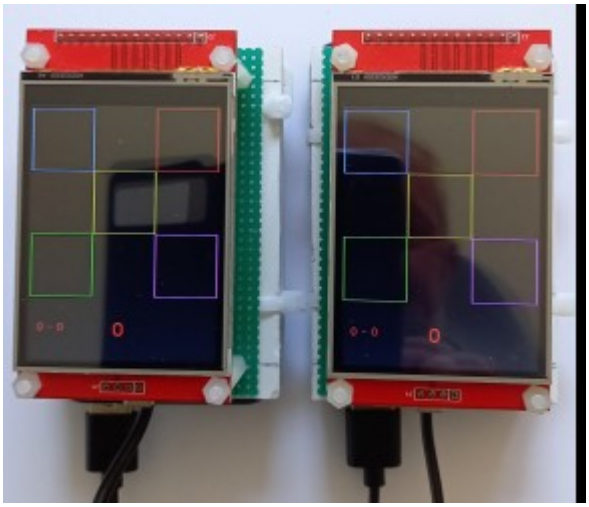
Middle
own points again

Right in green
which letter you are playing with

Yellow bar indicates whose move it is.

A correct combination is also shown here in blue.
New game after 5 seconds

Memory game



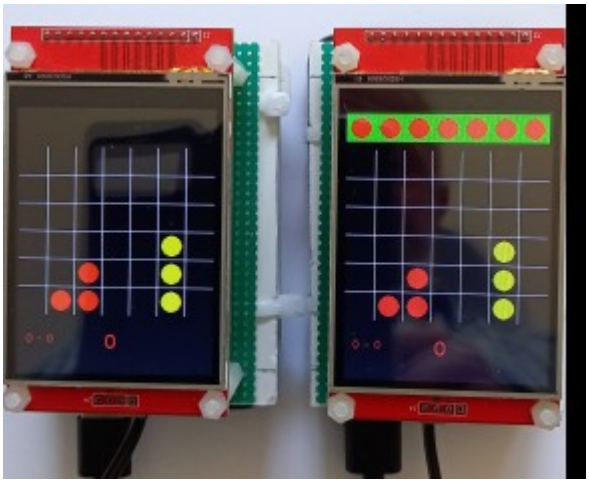
replay the colors shown by the ESP32 in the correct order.

Game starts with 3 colors.

Both players get the same colors.

If both players reach 5 points or a multiple of 5 points, 1 color is added, maximum is 15 colors.

Four In A Row, falling coins



Drop a coin by touching a red or yellow coin at the top of the bar.

Correct combination is shown in blue.
A new game starts after 5 seconds.

Lots of fun,
regards,
thieu-b55