Using VIDI-X Microcomputers in High School and College Education

Branko Balon* , Aleksandar Skendžić* and Milenko Simić**
*College for Information Technologies, Zagreb, Croatia
**The First Technical School Tesla, Zagreb, Croatia
branko.balon@vsite.hr, aleksandar.skendzic@vsite.hr, milenko.simic@skole.hr

Abstract **- This paper describes the application of the VIDI-X development microcomputer for educational purposes. VIDI-X is a platform developed entirely in Croatia and another great way to popularize STEM and learn new skills. VIDI-X is based on a dual-core ESP 32 processor and comes with a temperature sensor, a microphone, and a 2.8" touch screen. The possibilities of connection and application are enormous considering the diverse communication possibilities (Wi-Fi, Bluetooth, IR). VIDI-X is a significantly more powerful platform than Arduino, and unlike the Raspberry Pi computer, it does not need an operating system, and it can be programmed on a Windows, Linux, or Mac OS computer. It is possible to work in Python, C, C++, JavaScript, and other popular programming languages. Our students first installed the Arduino IDE on their laptops, and then connected the laptops to VIDI-X, and a series of exercises followed that combine the use of software and hardware. Our experience with this platform in the undergraduate computing curricula, at both the high school and college levels is presented here.**

Keywords - VIDI-X, ESP 32, STEM, Education,

I. INTRODUCTION

VIDI Project X was presented on December 5, 2019 in Croatia at conference for innovative projects awards powered by VIDI magazine. The whole project was developed by the VIDI media company in cooperation with a number of Croatian technology companies that financed the project.

VIDI-X is a microcomputer board that in its vision combines the "dream" and "realize" phases of the project, which we believe also requires innovation. The definition could be many, depending on the purpose you use it for. On VIDI-X microcomputer it is possible to run the original Doom or Duke Nukem in the emulator. You can connect additional sensors and manage the smart home system, you can use its Wi-Fi and Bluetooth communication capabilities to connect to an external system.

VIDI-X makes good use of the capabilities of the powerful dual-core ESP32 processor, so it can serve as a multifunctional microcomputer or as a development board, as an efficient HUB for a cluster of sensors or for some completely new purpose that is expected from the community that will gather around it to design.

Why did we choose VIDI-X microcomputer for education purpose at both the high school and college level? We wanted more powerfull platform, totaly new and completelly developed in our country - Croatia. We allready have experience with other widely used microcomputers boards like Arduino and Raspberry Pi microcomputers in education [4][6].

VIDI-X is a significantly more powerful platform than Arduino, and unlike a Raspberry Pi computer, it does not need an operating system. To run, a power source is required via a 3xAA battery or via USB cable, and it can be programmed on a computer with Windows, Linux or Mac OS system. Since its introduction, VIDI-X soon became very popular in Croatia also in the education, especially in STEM field [1] [7].

In II. chapter VIDI-X hardware and programming environment is described in detail. The examples of VIDI-X projects, that were introduced at laboratory exercise course in High school Tesla in Zagreb are shown in III. Chapter. The VIDI-X micro server project, that was used for demonstration in the *Computer Network course* at the College for Information Technologies - VSITE in Zagreb is described in last chapter of this paper.

II. VIDI-X HARDWARE AND PROGRAMMING

The VIDI-X development microcomputer is based on an ESP32 processor, a 2.8" integrated screen, a light sensor, an IR sensor, a temperature sensor, integrated speaker and the possibility of Wi-Fi, IrDA and Bluetooth connectivity.

Here is a quick overview of main parts od VIDI-X microcomputer The parts are numbered from 1 to 21 and are indentified on the front and the back side of VIDI-X as shown in Figure 1. and Figure 2.

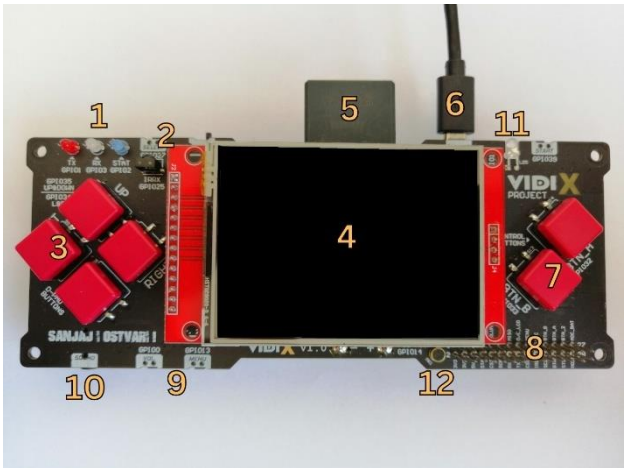

Figure 1. VIDI-X Front side

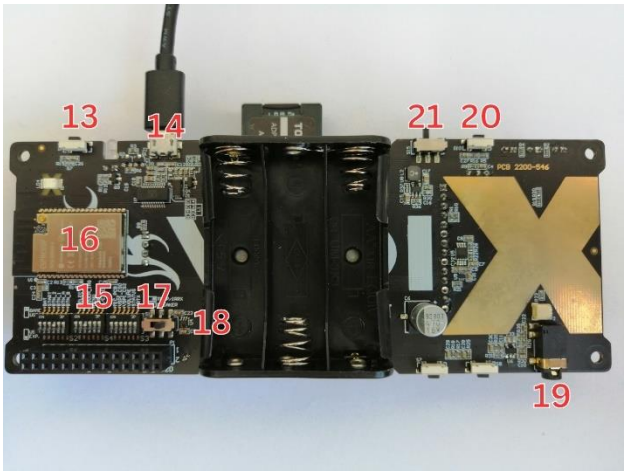

Figure 2. VIDI-X Back side

1. Red, white and blue LEDs - are control lights so we can visually check whether there is activity on the GPIO1, GPIO2 and GPIO3 pins. We mark GPIO2 as STATUS while the other two show serial TX and RX.

2. InfraRed receiver - will enable reception of ESP32-WROVER-B infrared signals from various remote controls of your lines, TV or other VIDI X microcomputer. It is connected to GPIO25.

3. BUTTONS - Two buttons are connected to one pin. Up - down on GPIO35, and left - right on GPIO34

4. Touch LCD screen - The screen has a resolution of 320x240 pixels and is touch sensitive. The screen itself can be easily separated from the rest of the board if necessary. A pen is included for easier use.

5. SD card - will enable more permanent storage of collected data from various sensors

6 . Micro USB - Power from the USB connector and serial communication for programming

7. Buttons labeled A and B - They are connected to GPIO32 and GPIO33 as marked on the board itself

8. Expansion header - it is available with a female version from the back and a male version from the front. On the front, next to the male pins, their markings are written for easier use

9. Volume and Menu - Volume and Menu buttons are connected to GPIO0 and GPIO13

10. Combo audio connector - The four-pole 3.5 mm audio connector for external microphone and headphones

11. InfraRed transmitter - connected to GPIO15 enable the transmission of infrared signals

12. Microphone - The microphone is connected to GPIO14 which serves as an audio input.

13. Start - The start button is connected to GPIO39

14. Micro USB - Power from the USB connector and serial communication for programming

15. Expansion header - In order to use the 28-pin expansion header, it is necessary to switch the microswitches next to it to "USE EXP." position. If they are in the "GAME USE" position, it is possible to use the buttons and sensors on the VIDI-X

16. ESP32-WROVER-B - VIDI-X microcomputer is based on the ESP32 module with a 32-bit dual-core processor that has 600 DMIPS (Dhrystone benchmark). 600 DMIPS means that the processor is capable of processing 600 iterations of the main loop code per second.

17. Switch - With this switch we determine whether we want to use the audio amplifier connected to GPIO25 and GPIO26 to have sound, or in another position the InfraRed receiver and temperature sensor. The temperature sensor is connected to GPIO26 while the InfraRed receiver is connected to GPIO25

18. Temperature sensor - MCP9700AT-E/LT has a measurement range from -40°C to 150°C with an accuracy of ±2°C

19. Speaker connector - This connector connects the speaker

20. Select - The Select button is connected to GPIO27

21. On/Off switch

VIDI-X also has three buses, UART, SPI and I2C, which it uses for direct communication with sensors, a computer or some other micro-controllers. Direct communication with other devices is achieved through the serial UART bus and General Purpose Input/Output (GPIO) connection pins as shown in Figure 3.

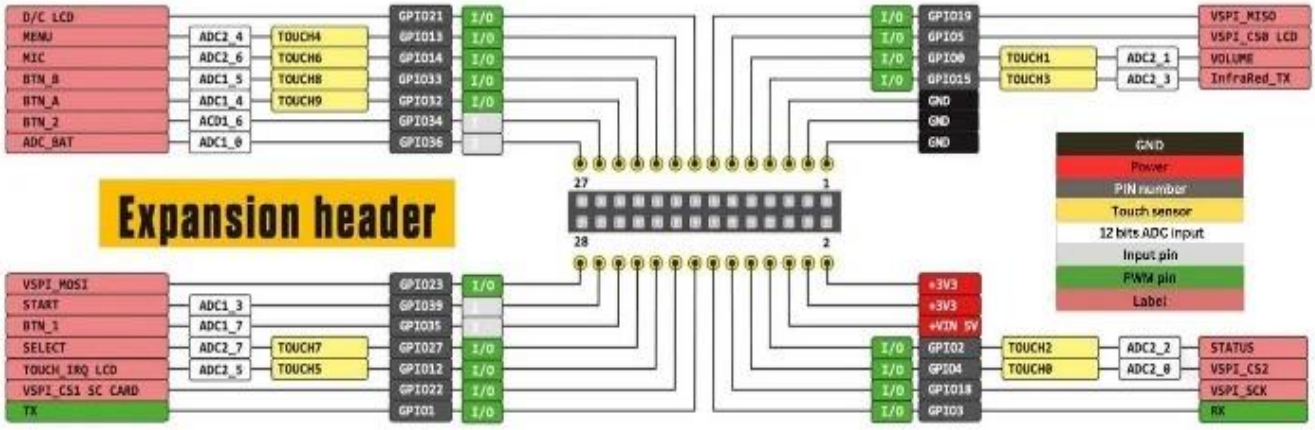

Figure 3. GPIO connection Source: https://VIDI-x.org/

Basic technical specifications of the VIDI-X processor module ESP32 are shown in Table I.

TABLE I. BASIC TECHICAL DATA - ESP32 MODULE
Source: https://VIDI-x.org/

Producer:	Espressif Systems
Wi-Fi:	802.11 b/g/n, BT
IrDA	yes
Frequency:	2.4 GHz to 2.5 GHz
Data transfer:	150 Mb/s
Security:	WPA/WPA2/WPA2-Enter-prise/WPS
CPU:	Dual-core Tensilica Xtensa LX6 microprocessor @ 240 MHz
Memory:	520 KB SRAM, 8 MB SPI flash memory

The VIDI-X microcomputer supports multiple development environments such as Arduino IDE, Visual studio CODE and PictoBlox, and Windows and MacOSX platforms are supported [1]. The development environment used in the projects is the Arduino IDE (Integrated Development Environment), and it was used for writing/uploading to the VIDI-X microcomputer. The Arduino IDE development environment is also used for compatible Arduino microcontrollers [2] [5] . The Arduino IDE environment supports the C and C++ programming language with code structuring rules and the necessary libraries used to compile the ESP32 processor code.

III. VIDI-X PROJECTS IN HIGH SCHOOL TESLA

VIDI–X development microcomputers are used at laboratory exercise course at the First Technical School and computer technicians program. Laboratory exercises are organized in small groups (maximum 10 students). Exercises are divided into two main parts. The first part refers to getting started with VIDI–X development microcomputer and to become familiar with its design. In the second part of this laboratory exercises students do some practical electronic ojects. They combine various electronic components and write code with Arduino IDE (Integrated Development Environment) to control them.[3] [4] [5].

A. LED control on VIDI-X

VIDI-X has 3 control LEDs so that we can visually check whether there is activity on the GPIO01 (red), GPIO02 (blue) and GPIO03 (white) pins. The red and white LEDs induce serial communication and are not programmable. GPIO02 (blue) LED is status and we can program it. The designation GPIO represents a general purpose input/output. Here is the example of the code:

```
int LEDPin1 = 2;
void setup(){
 pinMode(LEDPin1, OUTPUT);
 digitalWrite(LEDPin1, LOW);
}
void loop(){
 digitalWrite(LEDPin1, HIGH);
 delay(1000);
 digitalWrite(LEDPin1, LOW);
 delay(1000);
}
```

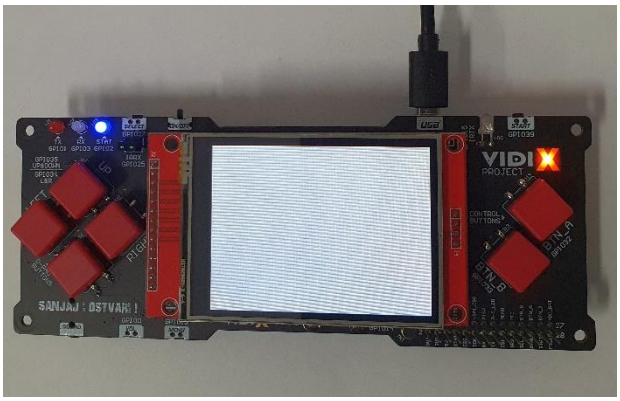

Figure 4. GPIO02 blue LED is programmable status light

B. *Temperature sensor management on VIDI-X*

As mentioned before on the VIDI-X board there is a temperature sensor MCP9700AT. It is located on the back of the board and is connected to GPIO26. In order to be able to use this sensor, the switch must be set to the TEMP/IRRX position. The temperature is read in 12-bit resolution, which means that it is possible to read 2^{12} = 4096 different values. The temperatutre readings are sent to serial port as shown in Figure 5.

Example of the code:

```
int PinTemp = 26;
int temp;
float tempV;
float tempC;
void setup() {
 pinMode(PinTemp, INPUT);
 Serial.begin(9600);
}
void loop() {
 temp = analogRead(PinTemp);
 tempV = map(temp, 0, 4095, 0, 3300);
 tempC = tempV / 10 - 40;
 Serial.print((String)"Temp in C = ");
 Serial.println(tempC);
 delay(1000);
}
```

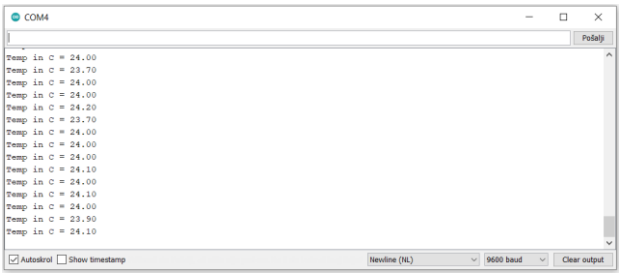

Figure 5. Temperature readings

C. *Servomotor control on VIDI-X*

Servomotors are DC motors that enable precise positioning of the motor shaft. Most hobby servomotors can rotate their axis by 180°, but there are also those that can rotate up to 270° or even 360°. In this project , the SG-90 motor is used, which can make 180°. In addition to the angle the shaft can make, servomotors differ physically in the size and construction of the gears (plastic or metal). They can be controlled analog via PWM signals or digitally via I/O signals. Most hobby servos are powered by 5V-6V, so the servo can be directly connected to the VIDI-X without an additional battery. However, if you want to connect more servomotors, you need to add an external power supply. In order to be able to manage the servo motors on the VIDI-X, it is necessary to download the ESP32Servo command library from the Internet by selecting Tools, then Manage Libraries, within the Arduino IDE interface. Here is the example of the code:

```
#include <ESP32Servo.h>

// create servo object to control a
   servo

Servo myservo;

int pos = 0;    // variable to store the
                   servo position

void setup() {
  myservo.attach(13);  // attaches the
     servo on pin 13 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) {
// goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);
    delay(15); // waits 15ms for the
        //servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1){
 // goes from 180 degrees to 0 degrees
    myservo.write(pos);
    delay(15); // waits 15ms for the
        //servo to reach the position
  }
}
```

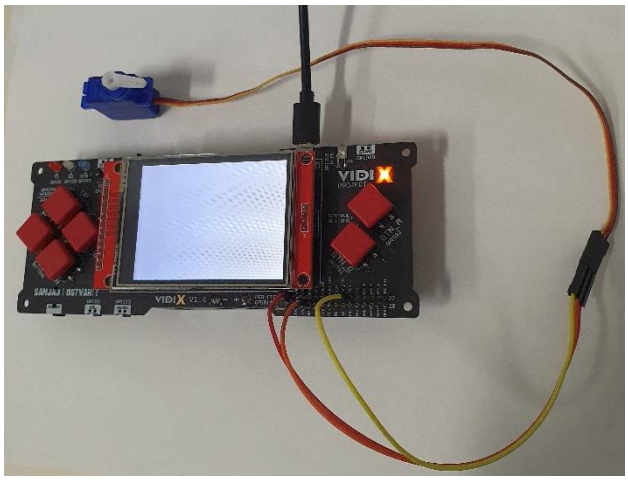

Figure 6. Servomotor controlled by VIDI-X

D. Controlling the TFT LCD on VIDI-X

The screen that comes with VIDI-X is a 2.8" ILI9341 SPI TFT LCD screen with a resolution of 320x240 pixels and is touch sensitive. The screen itself has many pins, some of which are intended for power supply, some for management, some for sending data that will be displayed on the screen, and some for receiving information if you write on the screen.

The ILI9341 driver is used to control the display, which must be additionally downloaded from the Internet and installed. To download and install it is necessary to select Manage Libraries in the Tools menu within the Arduino IDE. Here is the Example of the code:

```
#include <Adafruit_ILI9341.h>
#include <Adafruit_GFX.h>

// ILI9341 TFT LCD pin declaration
#define TFT_CS 5
#define TFT_DC 21
// create a screen object that will be
called tft
Adafruit_ILI9341 tft
    = Adafruit_ILI9341(TFT_CS, TFT_DC);

void setup() {
 tft.begin(); // screen initialization

 tft.setRotation(3); // set orientation
        // screen color
 tft.fillScreen(ILI9341_BLACK);
 tft.setTextColor(ILI9341_WHITE);
 tft.setCursor(45,105); position
 tft.setTextSize(5); // text size
 tft.print("VIDI - X"); // print text

 tft.setTextColor(ILI9341_RED);
 tft.setCursor(0,0); // position
 tft.setTextSize(5); // text size
 tft.print("TESLA"); // print text
 tft.setTextColor(ILI9341_BLUE);
 tft.setCursor(170,200); // position
 tft.setTextSize(5); // text size
 tft.print("VSITE"); // print text
}
void loop() {
}
```

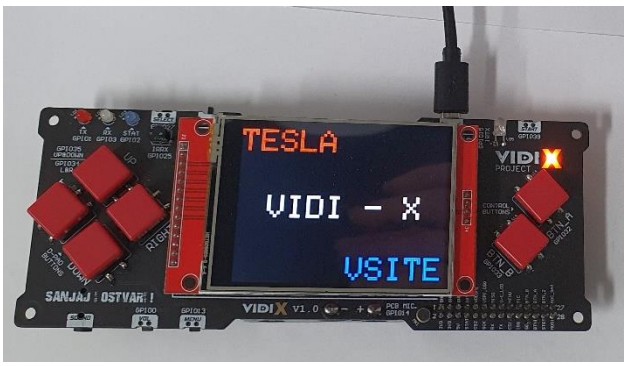

Figure 7. Displaying on VIDI-X TFT LCD

IV. VIDI-X VSITE COLLEGE PROJECTS

The VIDI-X micro server project was used for demonstration purposes in the *Computer Network course* at the College for Information Technologies VSITE in Zagreb. The basic idea of the project is the creation of a micro server with the possibility of network connection (soft Access Point) on which a simple HTML application is installed, through which the signal LED on the VIDI-X microcomputer is turned on/off. The network connection of the VIDI-X micro server is achieved via a wireless connection (SoftAP function), since there is no ethernet network connection on board. The goal of the project was to demonstrate the capabilities of the VIDI-X microcomputer in the role of a simplified micro server in a network environment. The project, in addition to the on-screen display of the network connection status, has been further extended with the functionality of an application that executes HTML code that turns on/off the integrated signal LED.

A. Development environment and used program libraries

The development environment used for the VIDI web server project is the Arduino IDE in version 1.8.10. [6]. The necessary program libraries were also used:

- WiFi.h - library for working with integrated Wi-Fi
- WebServer.h - server functionality library
- SPI.h - serial communication library
- gfxfont.h – library for exploiting the possibilities of TFT screens and printing on the screen
- AdaFruit_ILI9341.h - library for exploiting the possibilities of TFT screens and printing on the screen

B. *Basic network settings*

Network settings are specified for demonstration purposes. By calling the *IPAddress* function, the static IP address of the SoftAP server (192.168.1.1), the default Gateway and the name of the Wi-Fi network (Service Set Identifier - SSID) are defined. The defined network settings are used in the function *WiFi.softAPConfig*. The server listens to traffic on port 80 for http requests, which is accomplished with the *WebServer server(80)* function. Part of the program code of the used libraries and network settings is shown:

```
/* libraries needed */
#include <WiFi.h>
#include <WebServer.h>
#include <SPI.h>
#include <gfxfont.h>
#include <Adafruit_ILI9341.h>

/* SSID and password for wifi */
const char* ssid = "VIDI-X-PROJECT";
const char* password = "12345678";

/* Server IP address */
IPAddress local_ip(192,168,1,1);
IPAddress gateway(192,168,1,1);
IPAddress subnet(255,255,255,0);

/* Server listens on port 80 */
WebServer server(80);
```

C. *Wireless network settings – Soft Access Point*

Since the server has the possibility of wireless connection, the wireless network settings that enable this have also been defined. A wireless access point has been established, which enables the connection and reception of clients who authenticate. For this purpose, the following functions were used [7]:

- `server.begin();` – function that starts the server;
- `WiFi.softAP();` –function that defines VIDI-X as an access point
- `WiFi.softAPConfig();`- configuration parameter of the softAP function, i.e. micro server access point
- `WiFi.softAPgetStationNum();` - a function that obtains the number of clients currently connected to the micro server - SoftAP;
- `server.handleClient();` - listens to incoming HTTP requests;
- `server.onNotFound(handleNotFound);` - when the client requests an unknown URL (different from "/", for example), the "handleNotFound" function is called;
- `server.handleClient();` - a function that listens for HTTP requests from clients.

Definition of the softAP function for wireless connection to the micro server:

```
WiFi.softAP(ssid, password);
WiFi.softAPConfig(local_ip,gateway,sub
net);
delay(100);
```

Part of the program code that defines the number of wireless clients connected to the access point (server):

```
 server.begin();
 Serial.println("HTTP server started");
}
void loop(){
 tft.setTextWrap(false);
 Serial.println("No. of connected
clients");
 Serial.println
         (WiFi.softAPgetStationNum());
 tft.setCursor(10, 130);
 tft.println
         (WiFi.softAPgetStationNum());
 delay(100);
 server.handleClient();
 if (LED1status)
 {digitalWrite (LED1pin, HIGH); }
 else
 {digitalWrite (LED1pin, LOW); }
}
```

D. *Defiinition of screen display*

The *gfxfont.h* and *AdaFruit_ILI9341.h* libraries were used to take advantage of the built-in TFT screen and screen printing. The network configuration parameters and the number of active connections are displayed on the screen.

For this purpose, the TFT_DC and TFT_CS pins are defined, to which the TFT screen of the VIDI-X microcomputer is connected. The font colors used in the server status screen are also defined. Font colors are determined by hexadecimal notation (e.g. WHITE – 0xFFFF). The added functionality of the micro server is to turn on/off the LED connected to PIN 2 controlled by a simple HTML application placed on the server. To fill the screen with black, the command tft.fillScreen(BLACK) was used, which resulted in a better contrast, i.e. white letters on a black background. The command *tft.setRotation* rotates the print on the screen to the appropriate position, depending on the user's needs. The command *tft.setCursor(10, 0)* defines the initial position of text printing on the screen along the X or Y axis, while the commands *tft.setTextColor* and *tft.setTextSize* define the

color of the text to be printed on the screen as well as the size of the text. [7].

The following is the program code for printing the settings of the VIDI-X micro server, and the printout on the screen is shown in Figure 8.

```
/* definition when LED is ON */
void handle_led1on(){
 LED1status = HIGH;
 Serial.println("GPIO2 Status: ON");
 server.send(200, "text/html",
            SendHTML(true));
 tft.fillScreen(BLACK);
 tft.setCursor(10, 0);
 tft.setTextColor(WHITE);
 tft.setTextSize(1);
 tft.print("Server IP address: ");
 tft.println(local_ip);
 tft.setCursor(10, 10);
 tft.print("Gateway: ");
 tft.println(gateway);
 tft.setCursor(10, 20);
 tft.print("Subnet mask: ");
 tft.println(subnet);
 tft.setCursor(10, 100);
 tft.setTextColor(GREEN);
 tft.setTextSize(2);
 tft.println("VIDI Project X Web
              server");
 tft.setTextSize(1);
 tft.setTextColor(WHITE);
```

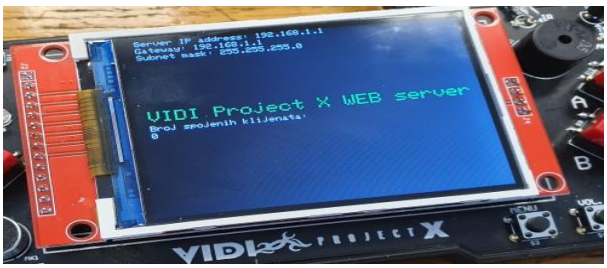

Figure 8. VIDI-X micro server. Display of server network settings and the number of connected wireless clients. Source: vidilab.com [7]

E. HTML Application and controll of LED lights

The VIDI-X micro server project has been expanded with additional functionality, which refers to the possibility of turning on/off the signal LED via an HTML application that is executed on the micro server. For this purpose, two functions are defined in the programming part of the code:

- `void handle_led1on()`
- `void handle_led1off()`

Functions *handle_ledon()* and *handle_ledoff()* **status code 200** (HTTP status) is sent, which corresponds to a confirmation response (OK). The content to be sent is "text / html", while at the end the *SendHTML()* function is called, which creates a dynamic HTML page that turns the LED light on and off. The *tft.printl* command prints the content on the screen. Figure 9. shows the interface of the HTML application running on the VIDI-X micro server.

Figure 9. Mobile application that turns on/off the VIDI-X LED
Source: vidilab.com [7]

V. CONCLUSION

The goal of the projects described in this paper was to demonstrate the capabilities of the VIDI-X microcomputer. Our experience with VIDI-X as an educational tool is exceptionally good. It is easy and fun for students to work with VIDI-X at both, the high school and college levels. All in all, it is another great way to popularize STEM and learning new skills. Anyone who is interested in more information will find them at the VIDI-x.org link, which presents hardware in more detail, the options that are significantly wider than these mentioned in this paper, explained examples of projects that can be made. Startup companies, hobbyists, researchers, as well as students receive a platform by which they can manage other devices, develop their own projects or use it as part of existing solutions.

REFERENCES

[1] VIDI-X, [Online] Available: https://VIDI-x.org/VIDI-x-razvojno-okruzenje/ [Accessed: 20. February 2023.]

[2] https://docs.arduino.cc/software/ide-v1/tutorials/Windows [Accessed: 18. February 2023.]

[3] El-Abd, Mohammed, " A Review of Embedded Systems Education in the Arduino Age: Lessons Learned and Future Directions" International Journal of Engineering Pedagogy (iJEP). 7. 79-93. 10.3991/ijep.v7i2.6845, 2017.

[4] B. Balon and M. Simić, "Using Raspberry Pi Computers in Education", 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, pp. 671-676, 2019.

[5] Novak, Milan & Kalova, Jana & Pech, Jiři, "Use of the Arduino Platform in Teaching Programming", 1-4. 10.1109/INFORINO.2018.8581788, 2018.

[6] B. Balon and M. Simić, "Arduino Platform as Learning Tool in High School and College Education" , 44th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, pp. 740-745. 2021.

[7] A. Skendžić "Vidi ProjectX #98,", VIDI Computer magazine No. 287, [Online] Available: https://www.vidilab.com/vidi-project-x/4753-vidi-project-x-web-server [Accessed: 20. February 2023.]