ADVANCES IN CYBER-PHYSICAL SYSTEMS
Vol. 7, No. 1, 2022

SMART INFORMATION BOARD BASED ON THE PSOC 6 MCU

Yurii Lozynskyi, Taras Torubka

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Author's *e-mail: taras.v.torubka@lpnu.ua*

https://doi.org/10.23939/acps2021.__.___

Submitted on 10.04.2022

***Abstract*: Nowadays, the development of electronics and related technologies is very fast. An extremely significant role in the life of moder society is took by information output devices, the development and modernization of which is an urgent task. Using the idea regarding wireless control of remote devices, an information board with remote control was developed. This article provides a reader with the development of universal information board which customers will be able to assemble for the dimensions that they need. The created device displays text news from websites, displays time and important messages. For the convenience of updating information and automation - the board should have WiFi with Internet connection. The information board is a great way to receive news quickly, without using other means to obtain information. The information board is designed on the CY8C624ABZI-D44 microcontroller. Also, important structural elements of this board are: buffer chip - SN74ALS244CN, LED matrix control chip - MAX7219, LED matrix - 1088AS and WiFi module - LBEE5KL1DX. The functional diagram of the module 1088AS and MAX7219 connected to PSoC 6 has been designed.**

***Index Terms*: server, HTML, client, XML file, WiFi, PSoC 6, microcontroller, display, data, MCU ARM, Quad SPI, NOR Flesh Memory, NT, RTOS.**

INTRODUCTION

Our world is changing every second and it is becoming more technological and modern. This way, a new reality needs absolutely up-to-date technological ideas. An extremely significant role in the life of modern society is taken by information output devices, the development, and modernization of which is an urgent task. It is not always convenient to take a smartphone out of your pocket to take a look at the news feed.

THE PURPOSE OF THE ARTICLE

The main purpose of this work is to develop a universal information board that customers will be able to assemble for the dimensions that they need. It is also important to simplify the interaction between the device and the user. Moreover, it is very significant to increase memory with PSoC 6 on Cortex-M4 in 3,9 times. Also, to increase MCU frequency in 5 times. Also, using the S25HS256T will increase the speed to 83 MBps, which will be 4 times more than the usual SPI whose speed reaches 21 MBps.This information board is able to join any web service to download information by itself. Moreover, it should have a convenient web interface to manage the system. Also, the information board has the ability to play audio signals.

REVIEW OF LITERATURE SOURCES

Source [1] shows the datasheet of the CY8C624ABZI-D44 MCU. It is a combination of a dual CPU microcontroller with low-power flash technology, digital programmable logic, high-performance analog-to-digital conversion and standard communication and timing peripherals.

Source [2] desribes the LED matrix control chip – MAX7219.

Source [3] shows the datasheet of LED matrix – 1088AS.

Source [4] presents the datasheet of the WIFI module – CYW4343W.

Source [5] provides the information regarding Transmission Control Protocol and Transport Layer Security that are among the most essential protocols in today's Internet.

Source [6] provides the information concerning the Universal Asynchronous Reciever-Transmitter that is one of the mosed used device-to-device communication protocols.

Source [7] desribes a generic description methodology for MCUs. It presents a thorough analysis of the structural aspects of an MCU, the various features of MCU peripherals and initialization sequences.

Source [8] provides the information regarding the WebSocket communication method in Internet of Things systems that are based on ESP32 microcontroller.

Source [9] describes the NOR Flash Memory.

Source [10] provides the information concerning the markup language – XML.

Source [11] shows the various approeches of usage of Pulse Width Modulation.

Source [12] desribes the Network Time Protocol.

Source [13] provides the information regarding the Real-time operating system. It focuses on the hardware-independent model of the context switch within a multi-tasking RTOS.

Source [14] provides the information regarding Wi-Fi Based Scrolling Digital Display With RTC using Arduino.

Source [15] shows the datasheet of ESP32.

Source [16] provides the main information regarding the ATMEGA328.

SYSTEM COMPONENTS

The system will be controlled by Infineon's CY8CMOD-062-4343W kit. The kit includes a microcontroller CY8C624ABZI-S2D44 and will be paired with a module for wireless networks CYW4343W. You also need to store information downloaded from web services somewhere - NOR flash memory S25HL512T will be used for this. The implementation of the display provides 2 ways:

1) monochrome display on matrices 1088AS of one color and chips decoder MAX7219;

2) color display on LEDs ws2812b.

There are also auxiliary components such as a photoresistor to automatically adjust the brightness of the display and speakers to play alarm sounds or music.

ANALYSIS OF RECENT DEVELOPMENTS

It will be good to compare my device with other solutions of this device that have been developed before.

The solutions described are mostly based on the Atmega328 microcontroller, which is significantly inferior to the CY8C624ABZI-S2D44. Table 1 shows the comparison beetwen my device and other.

As can be seen from the table, the new implementation of the information board on PSoC 6 is better than those already on the market. This is an alternative that gives more opportunities for the same money.

HARDWARE PART

PSoC 6 - Infineon microcontrollers are not as popular for different DIY as other companies such as STMicroelectronics (STM32, STM Nucleo) or Atmel (Arduino), but are powerful and versatile enough to create more complex projects based on them. CY8C624ABZI-S2D44 - 32-bit Dual CPU Subsystem. Which includes the main core Arm Cortex - M4F with an operating frequency of 150 MHz and the auxiliary core Arm Cortex - M0+ with an operating frequency of 100 MHz. The microcontroller can support a large number of peripheral interfaces such as SPI, UART, and I2C. In total, up to 13 such interfaces can work simultaneously, in addition, these interfaces are not strictly tied to GPIO and can work asynchronously. To communicate quickly with external memory, it is possible to use QUAD SPI. Two asynchronous SPIs were used to implement the first prototype. There is a 12-bit SAR ADC that can work in two different modes: differential and single-ended, a total of 16 channels can work. Let's not forget about such important things as counters, timers, and PWM of which 32 pieces in this MCU (Eight - 32 Bit and Twenty-Four - 16 bit) will be needed to properly display time on the dashboard and to timely connect and update the information to be displayed on the information board. Also, a big advantage of this microcontroller is a large amount of flash memory for programs which is 2048 KB, it will keep the web page in flash memory, which will be very appropriate in this system. Infineon also has CapSense technology. The second reason why CY8CMOD-062-4343W was chosen is the CYW4343W single-core chip that supports a single-band 2.4 GHz WIFI 4 and Bluetooth 5.1. The microcontroller requires a voltage of 1.7 V to 3.6 V [1], [7]. Fig. 1 shows the block diagram of CY8CPROTO-062-4343W.

Table 1

Comparison of Information Board on PSOC6 and BL21RGB WIFI,Neurosun,AL-QURAISH

Information board on Psoc6	BL21RGB WIFI,Neurosun,AL-QURAISH
The device is configured through a user-friendly web interface, which means that you do not need additional programs to work with the board, which is not convenient.	There are various cases: settings are not possible because they were not provided from the beginning, you need to download and install additional programs on the control device, settings by changing the configuration files.
Output of text and images specified by the user, as well as automatic download of news from web resources that the user chooses for himself.	Output of text and images specified by the user.
Monochrome image for 1088AS display and color for WS2812B.	Monochrome or color.
Modular design that will allow the user to create their own information board for his needs.	Constant sizes and shapes.
Receiving data from the user and downloading from web pages can be done via WIFI or via GSM.	Data is received by reading the memory card in the board, either via WIFI or via GSN.
More powerful MCU that allows you to create large displays.	Low-power MCU

Also a few words about the flash memory S25HL512T for which news will be stored. NOR flash memory manufactured by Cypress has a capacity of 512 MB [9]. Data transfer can be carried out in 3 different modes of SPI operation, depending on the data transfer mode the speed will also change. For example, when transmitting via Quad SPI, the speed may be 83 MBps, and when transmitting via SPI, the speed will be 21 MBps. The memory requires a voltage of 1.7 V to 3.6 V.

Using Real Time Operation System which allows you to speed up the system by responding to events that occur, ie you do not need to wait for a response from the user or server, and during this time you can perform some other operations. Also SPI interfaces that work at the hardware level and can work asynchronously. All this removes the load on the system, which allows you to direct computing power to other processes. But even without all these advantages in the software field, PSoC 6 on Cortex-M4 simply has more memory and 3-4 times higher MCU frequency.

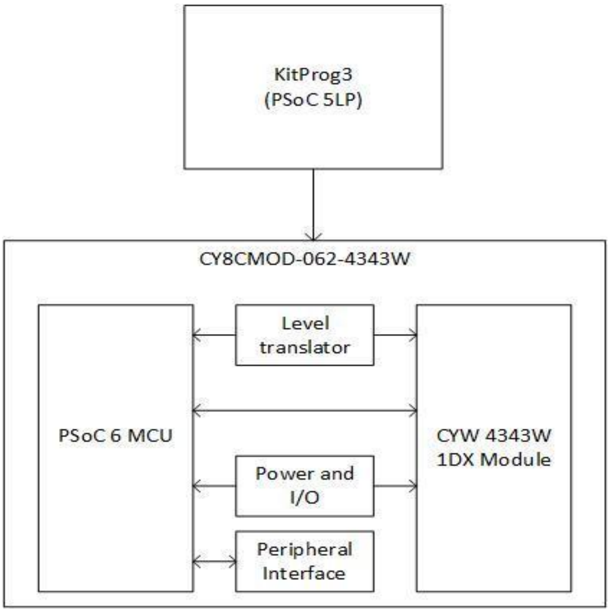

Fig. 1. Block diagram of CY8CPROTO-062-4343W

And now to the examples that already exist on the market, they are all based on microcoetrollers Atmel48/88/168/328 for comparison, let's take the best version of it ATMEGA328 (Arduino UNO, NANO) [16], also possible options on ESP32 [15], which works a little better. We are interested in flash memory where we will download the program, because in addition to the program itself, we will also keep a web page for the configuration server. Which can take up a lot of memory if you use complex scripts.

M_{PSoC6} shows the amount memory in PSoC6(KB) M_{ATMEGA} is memory amount in ATMEGA328(KB), M_{esp32} – is memory amount in ESP32(KB), k_{MPA} shows how many times less memory ATMEGA has then PSoC 6.

k_{MPE} shows how many times less memory ESP has then PSoC 6. CY8C624ABZI-S2D44 (PSoC 6) has 2 MB, which is quite a lot for a small MCU, ATMEGA328 has 32 KB, it turns out that

$$k_{MPA} = \frac{M_{PSoC\ 6}}{M_{ATMEGA}} = \frac{2048\,KB}{32KB} = 64,$$

ESP32 has 520 KB,

$$k_{MPE} = \frac{M_{PSoC\ 6}}{M_{esp\ 32}} = 3.9.$$

Although ESP and PSoC can be connected to external RAM. ESP will communicate with it through the SPI interface, of which it is not so many. We pass to SPI - for purity of comparison we will use only hardware instead of software. CY8C624ABZI-S2D44 (PSoC 6) - has 13 SCBs (serial communication blocks) are blocks that contain peripheral interfaces, they allow during the program to change their interfaces such as started with UART finished with I2C, so only 8 blocks support SPI and one separate QUAD SPI, this will allow you to easily connect 320 modules (MAX7219 and 1088AS) or for clarity 1 module has 64 LEDs.

Value m_{PSoC6} is the amount of modules which PSoC6 can control, m_{ATMEGA} is the amount of modules which ATMEGA328 can control, m_{esp32} is the amount of modules which ESP32 can control, number of LEDs in one module is 64.

Value P_{PSoC6} is the number of pixels which PSoC6 can control, P_{ATMEGA} is the number of pixels which ATMEGA328 can control, P_{esp32} is the number of pixels which ESP32 can control, then

$$P_{PSoC\,6} = m_{PSoC\,6} * 64 = 320 * 64 = 20480,$$

ATMEGA328 has 1 SPI, it will control 40 modules or $P_{ATMEGA} = m_{ATMEGA} * 64 = 40 * 64 = 2560$ pixels. ESP32 controls 160 modules and $P_{esp\,32} = m_{esp\,32} * 64 = 10240$ pixels, k_{PPA} shows how many times fewer pixels can control ATMEGA then PSoC 6, k_{PPE} shows how many times fewer pixels can control ESP32 then PSoC 6 $k_{PPA} = \frac{P_{PSoC\ 6}}{P_{ATMEGA}} = 8$. ESP32 has 4 SPI interfaces, $k_{PPE} = \frac{P_{PSoC\ 6}}{P_{esp\ 32}} = 2$.

Turning to the frequency of the MCU itself, PSoC 6 is dual-core on board, it has Cortex - M4 and Cortex - M0 + with frequencies of 150 MHz and 100 MHz, respectively. And we will equate the rest of the MCU with the weaker M0 core because it can also be programmed and it will be able to communicate with the main one via RTOS. k_{FPA} shows how many times less frequency ATMEGA328 has then PSoC6. k_{FPE} shows how many times less frequency ESP32 has then PSoC6 $k_{FPA} = \frac{f_{PSoC\ 6}}{f_{ATMEGA}} = 5$, $k_{FPE} = \frac{f_{PSoC\ 6}}{f_{esp\ 32}} = 2.5$, both lose significantly in speed.

Also the advantage of PSoC 6 is QUAD SPI and if you take a flash memory chip that supports this standard, for example S25HS256T speed will reach 83 MBps, and for normal SPI only 21 MBps sho S_{QSPI} is transfer speed of QSPI in MBps, S_{SPI} is transfer speed of SPI in MBps, k_S shows how many times QSPI is faster then SPI,

$$k_S = \frac{S_{QSPI}}{S_{SPI}} = 3.95$$

[14]

HARDWARE COMPONENTS FOR REALIZATION DISPLAY ON LED MATRIX

To implement the display of the information board, it is good to use 1088AS LED arrays, as the main idea of this project is the ability to design the display to the dimensions required by the user. 1088AS is a square matrix with 64 LEDs, ie 8x8 pixels that allows you to display almost any character from the ASCII table, the physical dimensions of such a matrix are 32x32 mm. Such a matrix can be controlled by PWM by supplying short pulses to the anode responsible for the column and to the cathode responsible for the rows, there are 16 pins at the bottom of the matrix for this. But, this method is somewhat impractical because it involves precious GPIOs on the microcontroller itself, although there are 102 we must not forget about the load on the system. Therefore, it will be good to use auxiliary chips for each matrix - decoders MAX7219. You can't do without it to build large displays. This chip allows you to control matrices using the SPI interface, which in turn greatly simplifies the work on hardware and software. MAX7219 from maxim integrated is a decoder that can work with 7-segment numeric LED displays up to 8 digitals or 64 individual LEDs which is equal to one 1088AS. This allows us to control each individual LED, the data rate is 10 MHz [2], [3], [4]. Fig. 2 shows the size and division of the command for MAX7219.

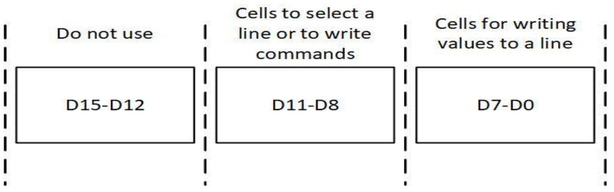

Fig. 2. Diagram of data amount

The DFPlayer Mini module is used to play the sounds and music needed to display important messages. Communication with the module is via the UART interface through which commands are sent. Sounds and music are recorded on a memory card that is installed in the DFPlayer mini module itself [6]. Fig. 3 shows the connection of the N-number of modules (not more than 10 per SPI channel).

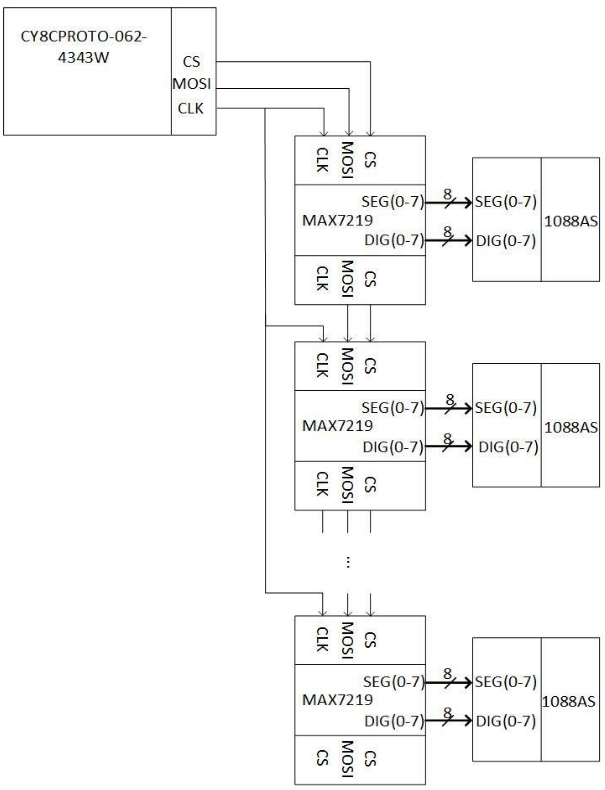

Fig. 3. Functional diagram of the module 1088AS and MAX7219 connected to PSoC 6

For programming one 1088AS matrix needed 2 bytes of information. In the first four bits we don't have any information this bits only needed to make the command equal two bytes, the next four bits used to match digits on a 7-segment display or to choose one of the eight rows in the matrix. Also this 4 bits use for a special commands like display intensity, decode mode (choose segment display or individual control 64 LEDs), display test and others. And the last eight bits uses for data like which led mast to on in row or if you use the command like display intensity this 8 bits will used to program the level of brightness. Another advantage of these decoders is the data transmission along the chain, which allows you to connect a large number of such modules to one SPI. For single module operation with MAX7219 and 1088AS, the voltage is 5 V, but the current will depend on the brightness level.

This diagram shows that the MOSI and CE signals are transmitted in series, and the CLK signal is transmitted in parallel. If the number of modules is more than 10, there is a high probability of delays in the MOSI signal, which can lead to incorrect reception of data from the SPI. The data for the MAX7219 is sent as follows - The first 2 bytes are transmitted sequentially to the last chip N in the chain, then the next 2 bytes are transmitted to the chip N-1 and so on until all are filled. The corresponding signal is then transmitted to the CS thereby including the corresponding LEDs or the execution of the corresponding commands. After signaling to the CS shift registers in the MAX7219 are cleaned and prepared for a new batch of data.

This way of implementing the display is quite cheap but it has disadvantages because such matrices can be only one color, LEDs in 1088AS are presented in 3 colors - blue, red, selenium. This monochrome reduces the possibilities of the information board.

IMPLEMENTING THE DISPLAY WITH THE WS2812B (COLORFUL)

To implement the color display, it is using LEDs WS2812B. WS2812B - RGB LEDs with custom shift registers. This makes it easy to communicate with them via PWM, and they also have the ability to transmit data in a chain as in the previously mentioned MAX7219, but with minor differences. If in MAX7219 the first set of data came to the last decoder. Then in the case of WS2812B the first data set remains in the first LED which after filling shift register, will send the following sets further on a chain. Fig. 4 is the diagram of data amount.

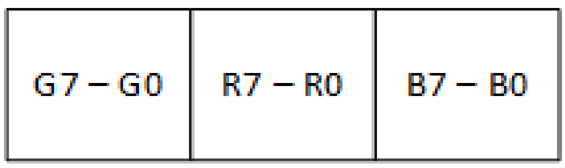

Fig. 4. Diagram of data amount (WS2812B)

Now let's talk more about what we will transmit for the WS2812B as this is what we will do with PWM. Since these RGB LEDs we have the ability to choose 16777216 colors, such a large number of colors available is that it accepts 8 Bits for each color - 8 Bits - red, 8 Bits - green and 8 Bits blue, ie in total for each LED needs to send 24 Bits of information this is shown in Fig 4. And now how will we communicate? Since PWM will be used, although you can also try to use SPI, but the version with

SPI will be more difficult to configure than PWM, so let's focus on the first. So just sending a logic 1 or 0 will not work, because for WS2812B 1 or 0 is a combination of low voltage and high voltage time. As can be seen from Fig. 5 in order to give a logic 0 you need to give a high signal level at 0.4 µs and a low signal level at 0.85 µs, and for logic 1 you need to give the opposite.

That is, the sum for 1 or 0 will require a cost of T=1.25 µs ± 600 ns. In order to reset the signal and clear the shift registers, a low signal level of more than 50 µs must be applied. Therefore, the proposed PWM interface is best suited for this. Calculate the fill level for PWM: the fill level for logic 1 $k = T_{on}/T = 0,64$, where T_{on} is the total period. The level of filling for the logical 0 by the same formula will be k = 32 %. You also need to know at what frequency the PWM should operate, to find out you need $f = 1/T = 800$ kHz. So now we have all the parameters to work with WS2812B using Pulse-width modulation.

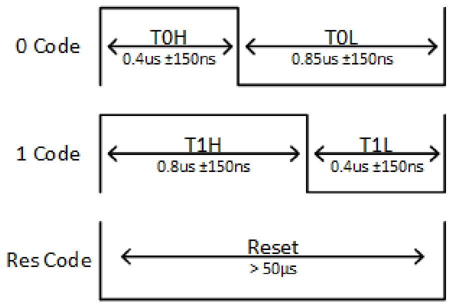

Fig. 5. Data transfer

Now consider the construction of the display itself, we need to create modules of 16x16 LEDs and combine them to the required dimensions, there are ready-made modules on a flexible basis, which further expands their scope. Fig. 6 shows the functional diagram of the connection CY8CPROTO-062-4343W to N number of modules WS2812B.

However, when connecting a large number of such modules, various problems related to the transfer of information via PWM may occur. For example, signal delays or various interferences, you can try to solve them with the help of buffer chips. Also, connecting all LEDs with one channel is undesirable because there is a high probability that if one LED burns out, the others that will go after that will stop working because they will not receive information.

There is another way to connect modules, each to connect a primitive microcontroller that supports PWM (for example ATtiny13 costs less than $ 1) - this microcontroller will be a driver and will operate PWM while the main device will communicate with him for example via UART [6].This may help solve some problems, but will definitely take the load off the PSoC 6. The main features of the WS2812B: the size of one LED 5x5mm is not very large, the operating voltage of 5 V current consumption will depend on the number of LEDs. Thanks to the capabilities of the Cortex-M4 core, the size of the display expansion can be increased 4 times.

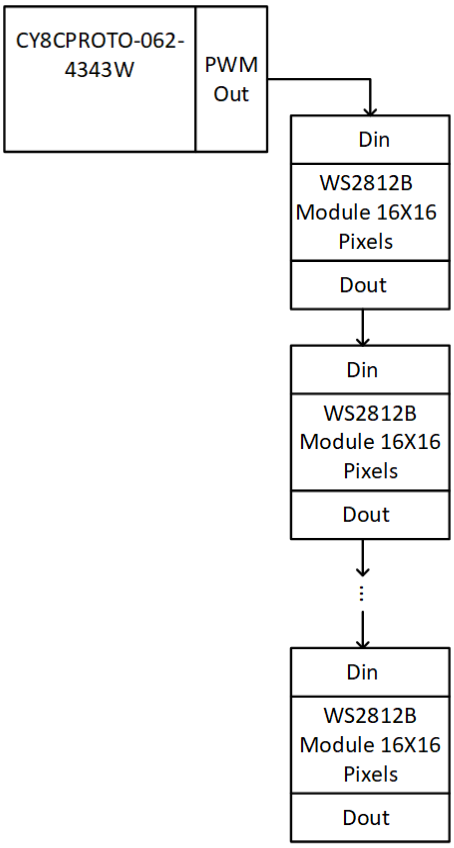

Fig. 6. Functional diagram of the module WS2812B connected to PSoC 6

SOFTWARE PART AND DEVELOPMENT OF THE FIRST PROTOTYPE

Libraries provided by Infineon (HAL - Hardware Abstraction Layer and PDL - Peripheral Driver Library), as well as MbedTLS were used to develop the software. The environment for developing the software was Modus ToolBox, the whole program is written in C and the Web page in HTML. Because PSoC 6 supports RTOS and the whole program is divided into tasks I offer and the article is divided into titles for each of the subtasks. But before that, a few words about the preparatory work of the program itself before the start of tasks [13]. Fig. 7 shows a block diagram of the information board.

Initially, PSoC 6 initializes the required interfaces such as SPI, WIFI, ADC and others. After that, the data for the WIFI network configuration is read from the flash memory. If they are not in memory then the microcontroller will try to create a local HTTP server and connect to the nearest non-password-free WIFI AP running at 2.4 GHz. The user needs to create a WIFI Access Point on their mobile device without a password and be as close as possible to the information board. After the PSoC connects to the network, the user needs to open a web browser and enter the IP address of the dashboard.

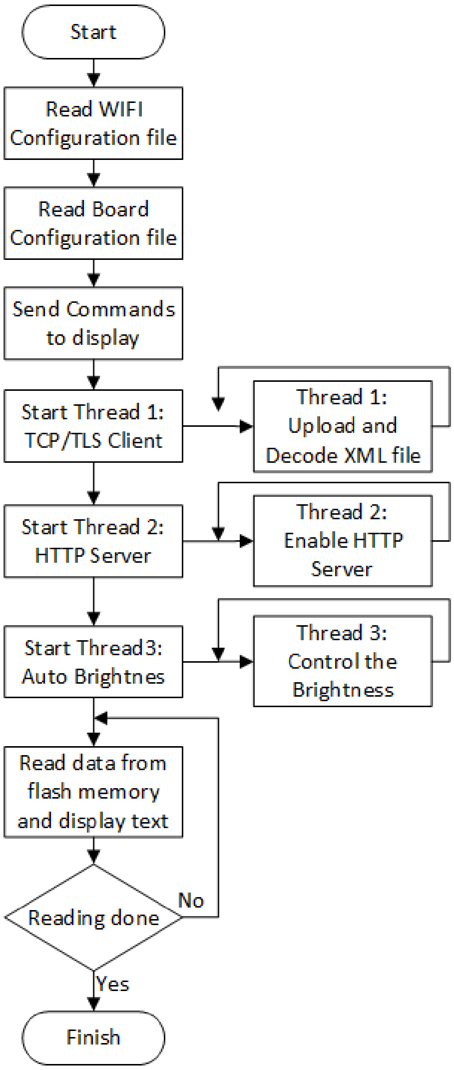

Fig. 7. Block diagram of the firmware

By entering the address, the user will see an HTML page where he can select the WIFI network to which the information board will connect and enter its password. After receiving the new settings, the information board will disconnect from the WIFI network created by the user and connect to the new one. Next in the scheme is reading the data for the display and the system itself. Data for the display includes fonts, text refresh rate, display brightness, auto brightness on, and other important settings. For the system, this is the last URL of the server to which the dashboard connects, the Root certificate to establish a secure HTTPS connection, and more. After receiving all the necessary information, the microcontroller will start adjusting the display. As mentioned earlier in this article, the MAX7219 chip (namely this device was created on them and 1088AS matrices) has special commands that will be sent to them [8].

THREAD 1 TCP/TLS CLIENT

This task starts almost from the very beginning of the program immediately after initializing the interfaces and reading all the information from memory. Fig. 8 shows the block diagram of the TCP/TLS Client tread.

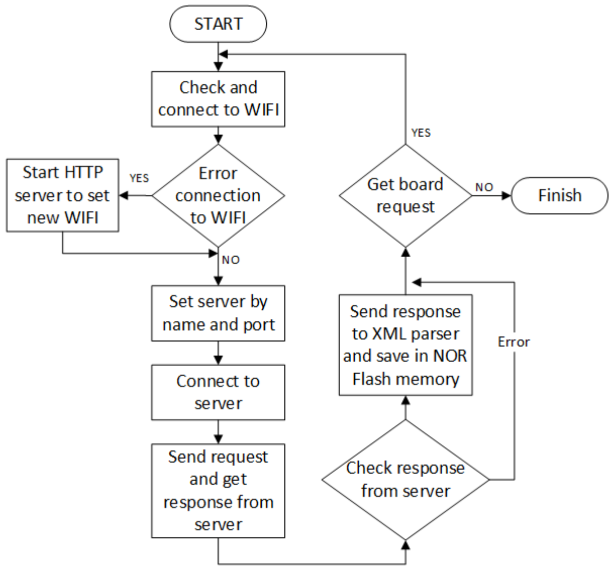

Fig. 8. Block diagram of TCP/TLS Client Thread

The TCP/TLS Client task is also the only task that does not work constantly, but is called by a timer that the user sets, after the task goes into standby mode. At each startup, the Client checks the WIFI connection and whether there is Internet access, if the connection cannot be established, the server mentioned earlier is turned on. If PSoC 6 successfully connects to the WIFI network, then it starts connecting to the Web Service via URL (for example, www.pravda.com.ua or www.radiosvoboda.org and others). After turning on the dashboard, the URL to connect to the server will be the one that was chosen before the shutdown. After all these long preparatory steps, after contacting the global server of a web resource and exchanging security certificates in case it is an HTTPS connection, the microcontroller starts communicating - in which it sends a GET request to the server to provide it with an XML file with RSS feed news, if the server likes everything and the user has specified the correct address where the file is located, the server will send a reply in which will be the text of this file.

A little explanation of what this XML file is is a file with an advanced markup language that allows you to easily parse the text in it. It is easy to do this because the whole text is divided into nodes, they can have different names in the file itself but usually it is <title>… </title> for the headline, and <description>… </description> for the description of the news itself. Although as it was said nodes can be different and so not all of them he can correctly recognize. After receiving this brief information about XML, you can proceed to the next block of the scheme, namely the parsing of XML text. To do this, a special library of functions was developed which quickly passes through the necessary nodes and extracts information from them. Then the received information will be saved on a flash memory chip, from which the text will then be displayed on the information board. After completing its work, the client is sent to standby mode until the next call on the timer. The system turned out to be quite complex, and many errors can occur in this chain of events, but most of them have been resolved so far. In addition, the user has no difficulty, he only needs the address of the resource from which he would like to receive information. Now we have dealt with the work of the first task, we move on to the next [5], [10].

THREAD 2 HTTP SERVER

This task is one of the features of this information board, as it allows the user to quickly and easily configure the device. You do not need any individual programs to connect to it or connect to it by wire and configure any files, all the user needs to join it is as simple as going to the website you need.

Fig. 9 shows a block diagram of the server, after the first operation of the task, it also checks the network connection as the first.

After successfully verifying the WIFI connection, the PSoC picks up the local HTTP Server and starts listening on the TCP port. If the user is connected, the microcontroller sends a POST request that contains an HTML page that opens like a regular website. At the same time, what the user needs to do to connect to the server - you need to be in the same WIFI network as the information board because the server is local (later a separate service will be added to help create a global server). Enter the IP address of the PSoC in the search bar of the browser, then the user will open a web page where he can configure the information board. What settings the user can make: 1) The speed of text on the display (because currently the scoreboard works as a scroll bar), 2) manually select the brightness level or select auto-brightness, 3) Choose the spacing between characters (this parameter is required because the user can add its own fonts and the interval will not always fit, plus the interval helps to make the text more readable at high speeds of its "running" on the display), 4) Space size, 5) How long will it take to call Thread 1 to update the news feed set in minutes), 6) Drop-down list where you can select already added web servers. It is also possible for the user to add new URLs from where the dashboard can download the news. But there is one unresolved issue at the moment - the microcontroller does not have the ability to generate a root security certificate to establish a secure connection, so the user must do it on their computer or phone. After setting up the board and adding a new news source, we move on to another information board service, namely "Important messages". This is the output of messages that the user will write, messages can be displayed only once or cyclically with a certain frequency that can also be set. You can also turn on audible alerts. After understanding all the features of the configuration server, we move back to the system. When the user configures everything he needs, he clicks the "download" button on the web page, after which the device will send a GET response to the microcontroller. The one who responds will save the new data and send a new web page with updated progenitors if the user is still connected. For example, if these are display settings - they will be saved in the settings file, a new website - a file with the website name will be created where all the necessary information such as URL, HTTP or HTTPS connection type and root certificate (if HTTPS) will be saved. Messages also have their own separate file. After all these manipulations, the server will continue to work. The task will be suspended only when Thread 1 is enabled to update the news. Now we have figured out how Thread 2 works and what it allows you to do, let's move on to the next one.

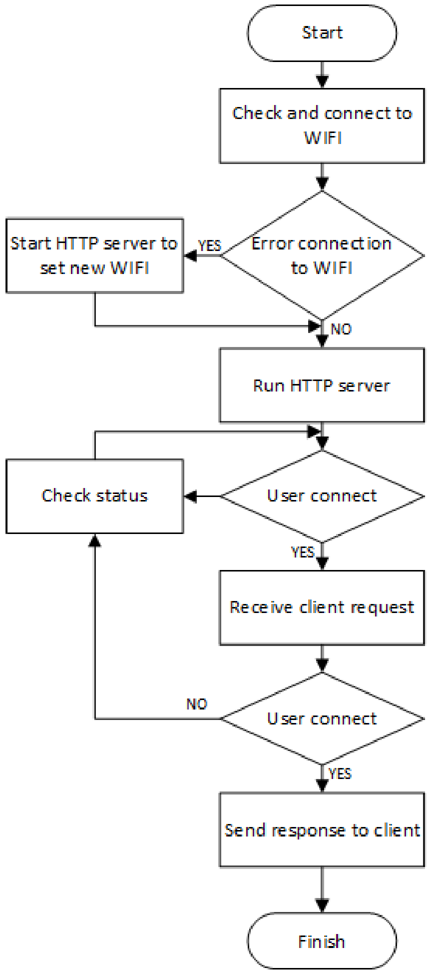

Fig. 9. Block diagram of HTTP Server Thread

THREAD 3 AUTO BRIGHTNESS

The simplest of the tasks, if the user has turned on the auto brightness - then the task will periodically go out of sleep mode to read the voltage from the photoresistor using the ADC. Auto-brightness has 15 levels, ie depending on the voltage level, the corresponding brightness value will be set.

From tasks to "important messages", time display and display operation.

Important Messages and Real-Time Output work together. Because the output of the first can be set to a specific hour, but you need to know what time it is in principle. It doesn't make sense to add a separate real-time chip because PSoC has a large number of counters, but you can't rely on them either, because turning off the board will save time. Therefore, the best decision was made - after turning on the microcontroller using the UDP network protocol connects to the real-time NTP server from where it will receive a specific date and time in the region [12]. This procedure is repeated at a frequency of 1 hour to avoid malfunctions of the clock. Therefore, the time on the information board will always be accurate.

DISPLAY TEXT

The text output process will be almost the same for both types of displays the only difference will be in the drivers. The prototype was created on the basis of MAX7219 and 1088AS, so the drivers that will be described are for them. Fig. 10 shows a diagram of sending text to the display.

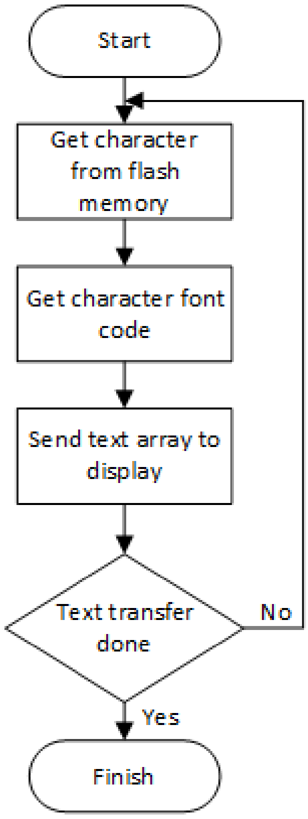

Fig. 10. Block diagram of text sending to display

Initially, the text that was downloaded Thread 1 will be read character by character, by the way the characters in Windows-1251 encoding because most web resources use it, but if the encoding is other than utf-16 then the characters are recoded to 1251 at the parsing stage in Thread 1. Next the symbol obtained from memory is compared with a two-dimensional array of fonts where the ordinal number of the cell corresponds to the symbol code.

By comparing numbers, the program receives a font code that will be converted by logical operations into a symbol on the display. For ease of understanding the program, the number of codes in one cell is not more than 5, which means that the maximum character width is 5 pixels, and the codes in the cell are 0 - 255, 8 bits, which means that the maximum character height is 8 pixels.

Prepared characters are written to a two-dimensional array of size 8x8 * N where N is the number of modules. After each sending of this array, the values in the cells are shifted by 1 using a logical shift, and a scroll bar effect is created on the display. The device currently described consists of 80 MAX7219 and 1088AS modules. To fill all 80 matrices with information, you need to send 8 * 80 * 2 = 1280 bytes of information. And that one point "ran" all board should be sent 80 * (8 * 80 * 2) = 102 400 bytes of information. The prototype itself is divided between two asynchronous SPIs of 40 modules each, this was done in order to eliminate the large signal delay. There was also a problem with Blockage of the fronts due to the capacity of the entrances. It was solved by using buffer chip SN74ALS244CN.

PROTOTYPE COMPONENTS

1) SN74ALS244CN - 1 Buffer chip;
2) MAX7219 – 80 LED matrix control chip;
3) 1088AS - 80 LED matrix;
4) CY8C624ABZI-D44 - 1 MCU;
5) CYW4343W - 1 WIFI module;
6) Power supply 5V 12A 60W;
7) DFPlayer mini - 1 mp3 player.

CONCLUSION

This article describes the process and all the necessary components to create an information board. Hardware components and software were described and analyzed. Also, the implementation of the display with the MAX7219 and 1088AS are described in detail and it is justified why this chip should be used. All three main tasks of the program are described in detail and the principle of their implementation is shown. Many diagrams have been presented to better understand the principle of the information board. Particular attention should be paid to the schematic diagram of the prototype information board.

Having developed this device, Increased memory thanks to PSoC 6 on Cortex-M4 3.9 times. Moreover, MCU frequency increased in 5 times compared to existing devices. Also, S25HS256T was used, which increased the speed to 83 MBps, which is 4 times higher than the usual SPI whose speed reaches 21 MBps.

REFERENCES

[1] Infineon.com (2021). *PSoC 6 MCU: CY8C62x8, CY8C62xA Datashee*t. [online] Available at: https://www.infineon.com/cms/en/product/microcontroller/32-bit-psoc-arm-cortex-microcontroller/psoc-6-32-bit-arm-cortex-m4-mcu/psoc-62/ (Accessed: 8 April 2022)

[2] Maximintegrated.com (2021). MAX7219/MAX7221 - Serially Interfaced, 8-Digit LED Display Drivers. [online] Available at: https://datasheets.maximintegrated.com/en/ds/MAX7219-MAX7221.pdf (Accessed: 8 April 2022)

[3] Alldatasheet.com (2021). *1088AS Datasheet (PDF) - List of Unclassifed Manufacturers.* [online] Available at: https://www.alldatasheet.com/datasheetpdf/pdf/1179268/ETC/1088AS.html (Accessed: 8 April 2022)

[4] Infineon.com (2020). *CYW4343W Single-Chip 802.11 b/g/n MAC/Baseband/Radio with Bluetooth 4.1.* [online] Available at: https://www.infineon.com/dgdl/Infineon-CYW4334WKUBGT-DataSheet-v11_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee1f02f680f (Accessed: 8 April 2022)

[5] Florentin Rochet, Emery Assogba, and Olivier Bonaventure. "TCPLS: Closely Integrating TCP and TLS", *ACM Digital Library,* 2020. doi: 10.1145/3422604.3425947.

[6] Eric Peña, Mary Grace Legaspi. (2020). UART: A Hardware Communication Protocol Understanding Universal Asynchronous Receiver/Transmitter. *Analog dialogue*. [Online]. *54(4).* Available at: https://www.analog.com/en/analog-dialogue/articles/uart-a-hardware-communication-protocol.html.

[7] Bhanu, G.P., Bai, Y., Tan, S.L., Chng, E.S., "A Generic MCU Description Methodology with Dependency Evaluation," 2009 International Conference on Signal Processing Systems, 2009, pp. 565-569, doi: 10.1109/ICSPS.2009.117.

[8] Mitrović, N., Đorđević, M., Veljković, S., Danković, D., "Implementation of WebSockets in ESP32 based IoT Systems," 2021 15th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS), 2021, pp. 261-264, doi: 10.1109/TELSIKS52058.2021.9606244.

[9] Brewer, J., Manzur G., "NOR Flash Stacked and SplitGate Memory Technology," in Nonvolatile Memory Technologies with Emphasis on Flash: A Comprehensive Guide to Understanding and Using Flash Memory Devices , IEEE, 2008, pp.179-222, doi: 10.1002/9780470181355.ch5.

[10] Seak, S.C., Siong, N.K., "A file-based implementation of XML encryption," 2011 Malaysian Conference in Software Engineering, 2011, pp. 418-422, doi: 10.1109/MySEC.2011.6140709.

[11] Zrilic, D.G., "Alternative Approach to Use of Pulse Width Modulation," 2006 World Automation Congress, 2006, pp. 1-5, doi: 10.1109/WAC.2006.376008.

[12] Chao, C., Huang, S., Hung, H., "Embedded System on NTP," 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, 2009, pp. 852-857, doi: 10.1109/ICCIT.2009.180.

[13] Gomes, R.M., Baunach, M., "A Model-Based Concept for RTOS Portability," 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA), 2018, pp. 1-6, doi: 10.1109/AICCSA.2018.8612862.

[14] Poudyal, S., "Wi-Fi Based Scrolling Digital Display With RTC using Arduino," 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT), 2019, pp. 199-202, doi: 10.1109/ICCT46177.2019.8969057.

[15] Alldatasheet.com (2020). *ESP32 Datasheet (PDF) - ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.* [online] Available at: https://pdf1.alldatasheet.com/datasheet-pdf/view/1243003/ESPRESSIF/ESP32.html

[16] Alldatasheet.com (2010). *ATMEGA328 Datasheet (PDF) - ATMEL Corporation.* [online] Available at: https://pdf1.alldatasheet.com/datasheet-pdf/view/392243/ATMEL/ATMEGA328.html

Yurii Lozynskyi, was born in 2001 in Lviv, Ukraine. In 2018 he started his stadies at Lviv Polytechnic National University at the Faculty of Computer Engineering. He won the first place in the Cypress Student Projects at 2020. He has worked at Infineon Semiconductor Manufacturing Company since 2021. His research interests include architecture in cyber-physical systems and pathfinding algorithms.

Torubka Taras, was born in Brodu, Ukraine, in 1986. From 2003 to 2008, he studied at the Lviv Polytechnic National University. Faculty: Computer engineering. Specialty: Special Computer systems. From 2008 to 2013 he was postgraduate student at the Lviv Polytechnic National University. From 2008 he has researching image processing: noise remove, image improve, object detection algoritms. Also, he implemented these algorithms on the microprocessors: Analog Devices BF533, Motorola DSP56F827.