ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 7, No. 1, 2022

MICROPROCESSOR SUBSYSTEM OF THE SMART HOUSE TO CONTROL THE MULTI-CHANNEL IRRIGATION OF THE ROOM PLANTS

Taras Borak, Dmytro Kushnir, Yaroslav Paramud

Lviv Polytechnic National University, 12, Bandera Str, Lviv, 79013, Ukraine.
Authors' *e-mail: taras.borak.mki.2020@lpnu.ua; dmytro.o.kushnir@lpnu.ua; yaroslav.s.paramud@lpnu.ua*

https://doi.org/10.23939/acps2022.__.___

Submitted on 19.04.2022

***Abstract*: This article develops the principles of building an intelligent home microprocessor subsystem to control the multi-channel irrigation of houseplants. The relevance of this topic has also been substantiated. Currently, there is a small number of devices in demand with a comfortable user interface and timer, which allows to adjust the watering at any time of day. The advantages over other available analogs and the need to create a customized system have been investigated. The developed structural-schematic diagram of the irrigation control system of houseplants based on the Arduino Nano microcontroller and a diagram of the algorithm of the subsystem has been proposed and given. As a result, there has been an example of the development of a subsystem that aims to improve and simplify the care of houseplants, which will save time and water resources.**

***Index Terms*: Computer System, Smart Home, Multi-channel Watering, Microprocessor.**

INTRODUCTION

Modern development of science and technology allows to automate and improve life in many areas of human activity, including in everyday life. Today, many devices save time and make people's lives at home and at work more comfortable, convenient, and cheaper. Houseplants are in almost every home. In this case, the condition of houseplants depends on timely watering and care. The main thing that plants need during your absence - water. Automatic watering systems are indispensable if the plants need to live independently for three weeks or more. The system of automatic watering of houseplants allows to provide various plants with water in time, to save both time, and water resources, however care of plants relates to difficulties of a choice of an optimum mode of watering. Thus, the need to develop intelligent irrigation systems determines the relevance of this work.

Currently, many automatic irrigation systems have many disadvantages, some do not save water well, while others use soil moisture sensors, which quickly oxidize and become unusable. Also, irrigation canals cannot supply water separately for each plant [1]. In smart home systems, built-in automatic watering systems for houseplants are rare, because these systems are usually made for cultivated plants [2]. Other smart home systems are installed in large homes, and they are not quite suitable for small spaces [3]. Before developing such systems, it is also necessary to investigate the conditions under which the system will exist, namely humidity, temperature and other indicators. convenient for plants in the room or the house [4].

The design of the new subsystem includes software development [5], structural study of the microcontroller to be used [6] and its interaction with other sensors [7]. It is also necessary to provide a user interface to adjust each plant's specific channels and watering time [8]. Furthermore, it is important to investigate the best connection of this interface to the microcontroller [9] and control channels that should exist independently, as well as to explore the most comfortable input device for future subsystem [11,12]. Such devices may be used as a unit in cyber-physical systems [13, 14].

A review of literature sources creates the relevance and feasibility of research on the development of a subsystem of automatic irrigation of houseplants

STATE OF THE PROBLEM

Most systems of automatic watering of houseplants have the following disadvantages:

- there is no possibility to adjust the water supply to each houseplant separately;
- lack of a convenient interface, where each houseplant is identified and the amount of water and the period of supply are indicated;
- not all systems are well enough suitable for small houseplants;
- high cost.

Many other systems use a humidity sensor, which causes inconvenience when a large number of plants, because it must be placed in the soil of each dew. Also, the sensors are rapidly oxidized in plants that require excessive watering, which leads to improper operation [1].

This work is devoted to the development of a subsystem of a smart home automatic watering of houseplants, which allows for timely watering of plants in accordance with the specified parameters.

FORMULATION OF THE PROBLEM

The paper is aimed at developing a subsystem of an intelligent home automatic watering of houseplants, which allows to water plants in time following particular parameters.

SOFTWARE FOR THE IMPLEMENTATION OF THE SMART HOME SUBSYSTEM

Intelligent home subsystems are automated control subsystems for various components of home infrastructure. With the help of special equipment, the system can recognize typical situations and respond to them by connecting specific components. In this case, the subsystem fully controls the operation of the device and prevents its irrational use. Thus, due to the "synergistic effect," the subsystems of an intelligent home allow to ensure the optimal use of the entire set of devices in the house. Moreover, this allows to create the most comfortable living conditions for people with the most economical consumption of resources [2-3].

The whole system of the "smart home" consists of three main components:

· Control point. Modern technologies allow to provide control of system components using the most various devices. It can be a simple switch, touch panel, or iPad. The remote control is provided by mobile phone, SMS, and other similar solutions.

· Central controller. This is, in fact, the brain of the whole system; all the information about the work of a module comes from. In addition, the central controller receives and processes commands received from the control points. With regard to the functions of the central controller, it became possible to control the harmonious operation of all components effectively.

· Executing devices. The irrigation system has a precise setting for when and when it is necessary to turn it on. Firstly, however, it is necessary to consider some nuances: the temperature of the water, the humidity of the air, and the ground, when it turns off. Furthermore, if we have several zones on the site that require humidification, we need to set a particular time for watering.

To set the appropriate level of irrigation, it is necessary to know in which zone this or that humidity is needed, so special sensors with a specific sensitivity will be installed. Then, due to a special controller, humidity, temperature, and, when necessary, water zone will be analyzed. Then, it is necessary to check the tank, a special compartment for filling water, if such is installed [4].

Fig. 1 shows a block diagram of a multi-channel irrigation control subsystem, which shows the connections between all subsystem components.

Multichannel is implemented by a switch, which can be like a standard decoder. If 16 channels are required, a 4-bit binary code must be applied to the control input.

In the first stage, the system is initialized with all stored values.

After starting the system, the mode of operation after starting the system describes the block diagram in Fig. 2.

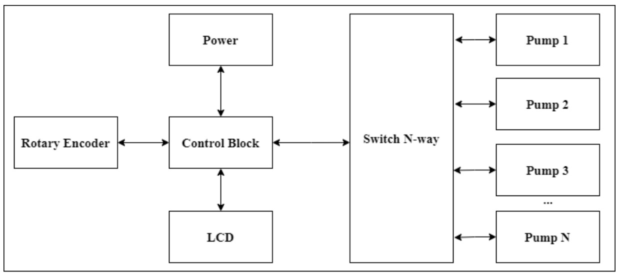


Fig. 1. Block diagram of the subsystem of multi-plant irrigation of plants

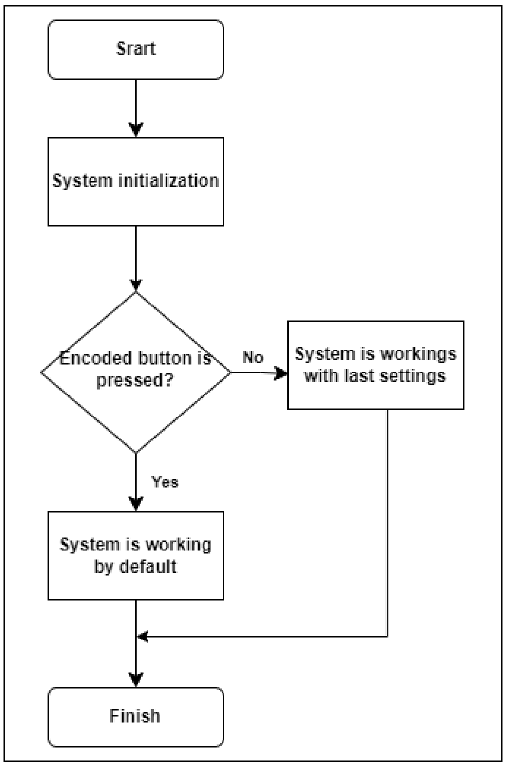

Srart
System initialization
Encoded button is pressed?
No
System is workings with last settings
Yes
System is working by default
Finish

Fig. 2. Block diagram of the settings reset

The successor step checks whether the encoder button was pressed during initialization. The reset – if this system is easy. If the encoder button has been pressed, all settings are reset to default values. However, if the button was not pressed, the system starts working with the latest settings

After the start-up and the settings for each water channel have been made, the subsystem is ready for operation. If the watering period and time for the channels are the same, then watering will take place sequentially, pausing the watering period. This feature ensures stable operation of the system and helps to avoid overvoltage. The block diagram of the subsystem after the settings is shown in Fig. 3.

The software implementation of this system is written in C ++ in the integrated environment Arduino IDE, because this environment has the following advantages:

- quick code filling into the Arduino board;
- in order to mark a specific port pin as input or output, it is enough to write the pinMode function (Port name, OUTPUT / INPUT);
- quick assignment of a port number to a certain variable;
- quick indication of port signals status (LOW or HIGH) [5].

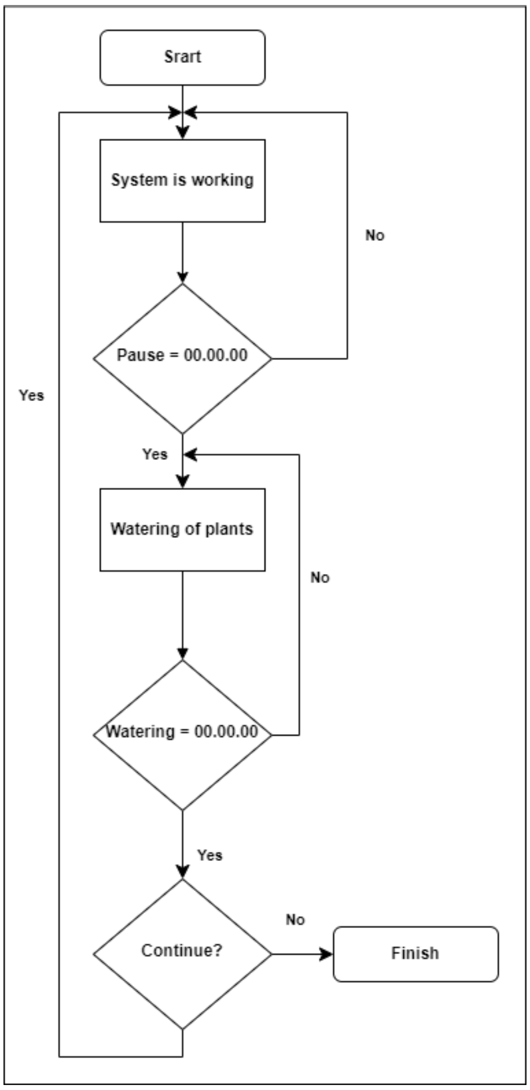


Fig. 3. Block diagram of the subsystem algorithm

The purpose and characteristics of the libraries that have been used below.

EEPROM: memory whose values are stored when the board is disconnected (like a small hard drive). This library allows to read and write these bytes. The library's purpose is also to support other standard data types: it currently implements write and read to int, long, float, and double.

LCD_1602_RU: the library allows to use Cyrillic when using LCDs connected by an I2C interface. Maximum it is possible to show 8 Cyrillic characters (e.g., W, D, Y, etc.). Characters identical to English (A, B, C, O, P, etc.) are used with the English character set. Additionally, the built-in feature displays the value of degrees Celsius. In-text programs, we must enter the UTF-8 code (Alt + 0176).

GyverEncoder: this library is designed to work with encoders, including KY-040 modules. The main functions are the following: working out of turn of the encoder, working out of the pressed turn, work with two types of encoders. Practice pressing or holding the button without rattling.

LiquidCrystal: this library allows the Arduino to control LiquidCrystal displays (LCDs) based on the Hitachi HD44780 (or compatible) chip, which can be found on most text LCDs. The library operates in 4- or 8-bit mode (i.e., using 4 or 8 rows of data compatible with RS, Enable, RW) [12].

HARDWARE IMPLEMENTATION OF THE SMART HOME SUBSYSTEM

Thus, for optimal watering of the plant, it is necessary to calculate two parameters that depend on the above factors: the amount of water that must provide to the plant; the time at which it is necessary to water the plant. In this case, a timer will be used that will indicate a specific time and frequency of watering for a particular plant. Therefore, during the system design, the following main components have been selected, the list of which is given below:

- Arduino NANO 328p;
- LCD1602 display;
- KY-040 module with button;
- I2C module;
- relay module for 4 channels with a capacity of 5V.

Arduino Nano is a small, full-featured board adapted for work with a mock-up board, which is based on the ATmega328 or Atmega168 microcontroller. It has the same functionality as the Arduino Duemilanove but it is smaller. It differs only in the absence of a power connector and operation via mini-USB. Arduino Nano is designed and manufactured by Gravitech.

Power in the Arduino Nano is via a mini-B USB connection, an external unstabilized 6-20V power supply, or a stabilized 5V power supply. The power supply with the highest voltage is selected automatically.

ATmega328 has 32 kilobytes (of which 2 kilobytes as well used by the bootloader). In addition, ATmega328 contains 2 kilobytes of SRAM and 1 kilobyte of EEPROM.

Each of 14 digital pins of the Arduino Nano can be used both as an input and as an output, using the functions pinMode (), digitalWrite () and digitalRead.

They operate at a voltage of 5 volts. Each output can pass a maximum current of 40 mA and has an internal resistor (default disabled) 20-50 kO.

A detailed board pinout is shown in Fig. 4 [12].

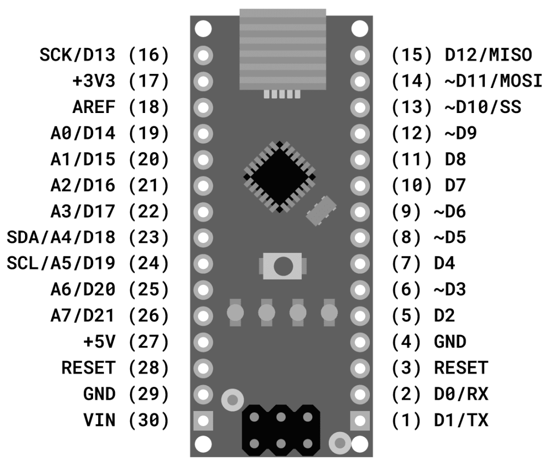


Fig. 4. Arduino Nano board pinout

Also, some outputs have special functions:

· serial port: 0 (RX) and 1 (TX). Outputs are used to receive (RX) and transmit (TX) serial data with TTL levels. These outputs are connected to the corresponding pins of the FTDI chip USB converter.

· External interrupts: 2 and 3. These outputs can be configured to trigger an interrupt at the front or at a pulse drop or by changing the level at the output;

· PWM: outputs 3, 5, 6, 9, 10, and 11.

· SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These outputs communicate via SPI;

· LED: 13. The built-in LED is connected to digital output 13. At a high level, the output LED lights up; when low – it goes out;

· I2C: A4 (SDA) and A5 (SCL). Maintain communication via I2C (TWI) by using the Wire library.

Arduino Nano is programmed using the Arduino IDE. ATmega328 on Arduino Nano already comes with a stitched bootloader, which allows to download new code to the controller without additional programmers. The bootloader works with the STK500 protocol [6-7].

LCD1602 is a liquid crystal display with I2C interface. The integration of the LCD display greatly facilitates the project's interactivity, allowing the user to read some of the output parameters directly. These values can be both plain text and numeric, such as temperature or pressure, or even the number of cycles performed by the Arduino. Pinout of LCD1062 is mentioned in Fig. 5 [8].

However, these displays have a small problem. When connected to a microcontroller (such as an Arduino), these displays require many PIN connections, occupying almost all available I / O, and leaving few outputs for any other devices and sensors. This problem was solved with regard to the connection on the I2C bus. The LCD1602 display has an integrated microchip that controls this type of connection, and then all input and output information is limited to only two PIN codes (without power supply).

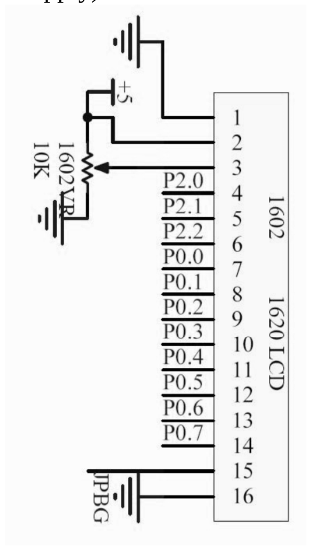


Fig. 5. LCD1602 pinout

I2C is a serial bus developed by Philips that uses two directional lines called SDA (Serial Data Line) and SCL (Serial Clock Line). Both must be connected by retractable resistors. Usage voltages are standard as 5V and 3.3V. The LCD1602 module consists of two parts. The backlight can be green or blue. With this display, we can see our settings, select a specific houseplant, and set the required time and frequency for watering [9].

The relay has three high voltage terminals (NC, C, and NO) that are connected to the pump. On the other side, three low voltage outputs (ground, power, and signal) connect to the Arduino microcontroller. A relay is a gateway that allows to connect electrical circuits with different parameters. The relay turns on or off external devices, in some way closing or opening a separate electrical network to which they are connected. With the help of Arduino and relays, we control the process of turning on or off our pumps by giving a command to close or open.

The designed subsystem used a normally closed configuration in which a high signal opens the switch and interrupts the current of 120-240 volts. A low signal closes the switch and allows current to flow from terminal C to terminal NC.

Therefore, a high signal interrupts the current [10].

Inside the KY-040 module, there are two switches and a separate sensor with three outputs (A, B, C). The first switch connects output A to output C, and the second connects output B to output C. In each fixed position of the sensor, both switches are either open or closed. Each click means to switch the state as follows: if both switches are closed, turning the axis clockwise or counterclockwise to one position will move both switches to open, and if both are open, turning the axis clockwise or counterclockwise to both positions to one position switches in the closed state. The module is designed so that a low logic level appears when the contacts are closed and high when the contacts are open. The low signal is generated by shorting contact C to the common wire, and zero is fed at this time to the output CLK and DT when the switch is closed. A high level is generated by supplying a supply voltage of 5 volts through a resistor. In this case, the outputs CLK and DT will be ones when the encoder contacts are open. Similarly, this encoder has a button, an integral part of it. If we press the button, the normally open contact of the button will close [11].

Because the LCD 1602 requires many microcontroller inputs, this number has been reduced by serial communication (I2C module), which sends data packets one after the other using only two inputs of our microcontroller, A5 and A4. First, the I2C module was connected to the LCD, as it is shown in Fig. 6. Next, connect the I2C module to the inputs of our microcontroller, connecting the power and ground inputs. Fig. 7 shows the connections between Arduino and I2C module. The SDA input of the I2C module is connected to the A4 input and the SCL input to the A5.

Each I2C bus consists of two signals: SCL and SDA. SCL is the clock signal, and SDA is the data signal.

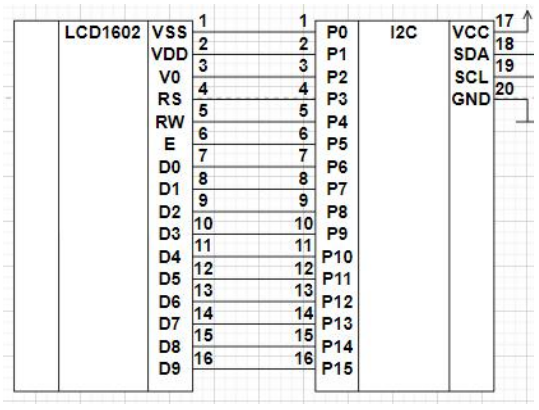

LCD1602
VSS VDD V0 RS RW E D0 D1 D2 D3 D4 D5 D6 D7 D8 D9
I2C
P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15
VCC SDA SCL GND

Fig. 6. Wiring diagram of the LCD1602 display with the I2C module

The 4-channel relay is connected to the microcontroller by connecting the signal inputs of the relay and the digital inputs of the microcontroller. The input of the relay is inverted, so a high level at the input excludes the coil, and a low – includes it. It can be noted that it is possible to increase the number of pumps. Take an 8- or 16-pin relay and connect it to other free digital and analog inputs of our microcontroller. 4-way relay is connected to Arduino in Fig. 8. A maximum of 14 pumps can be connected to the relay. Digital inputs D4, D5, D6, and D7 were used in the designed work.

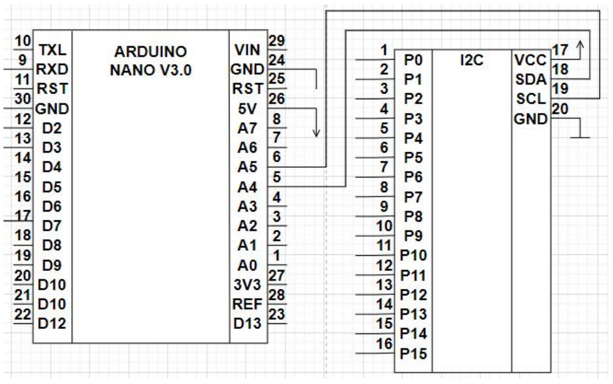

ARDUINO NANO V3.0
TXL RXD RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D10 D12
VIN GND RST 5V A7 A6 A5 A4 A3 A2 A1 A0 3V3 REF D13
I2C
P0 P1 P2 P3 P4 P5 P6 P7 P8 P9 P10 P11 P12 P13 P14 P15
VCC SDA SCL GND

Fig. 7. Wiring diagram of the Arduino Nano with the I2C module

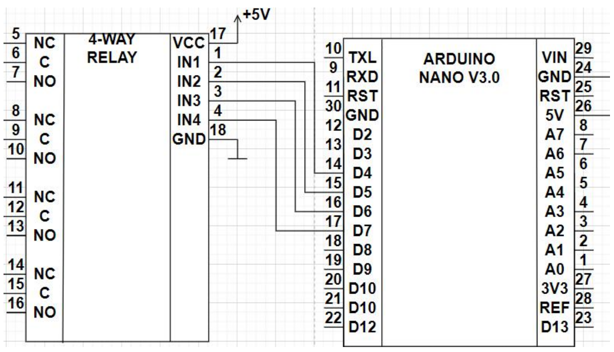

+5V
4-WAY RELAY
NC C NO
VCC IN1 IN2 IN3 IN4 GND
ARDUINO NANO V3.0
TXL RXD RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D10 D12
VIN GND RST 5V A7 A6 A5 A4 A3 A2 A1 A0 3V3 REF D13

Fig. 8. Wiring diagram of the Arduino Nano display with the Relay module

The connection of the encoder module is straightforward: power to power (GND and VCC), logic pins CLK, DT (clock pins of the encoder), and SW (pin of the button) to any pins of the Arduino (D or A). For round modules, the encoder pins are labeled as S1 and S2, and the button pin as Key is connected in the same way. The "direction" of its operation depends on the order of connecting the clock outputs of the encoder, but this can be corrected in the program. For example, the KY-040 encoder module was connected using the inputs of the microcontroller pins D0 (RXD), D2, and D3. An example of these connections is shown in Fig. 9. It is also possible to use other digital and analog inputs.

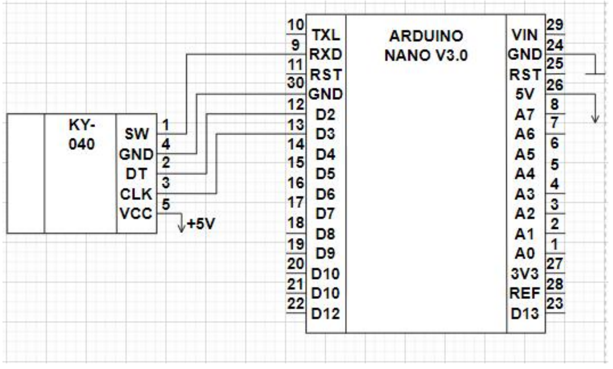

KY-040
SW GND DT CLK VCC
+5V
ARDUINO NANO V3.0
TXL RXD RST GND D2 D3 D4 D5 D6 D7 D8 D9 D10 D10 D12
VIN GND RST 5V A7 A6 A5 A4 A3 A2 A1 A0 3V3 REF D13

Fig. 9. Wiring diagram of the Arduino Nano display with KY-040 module

In the end, after all the subsystem elements were connected, the basic electrical circuit was designed, as it is shown in Fig. 10.

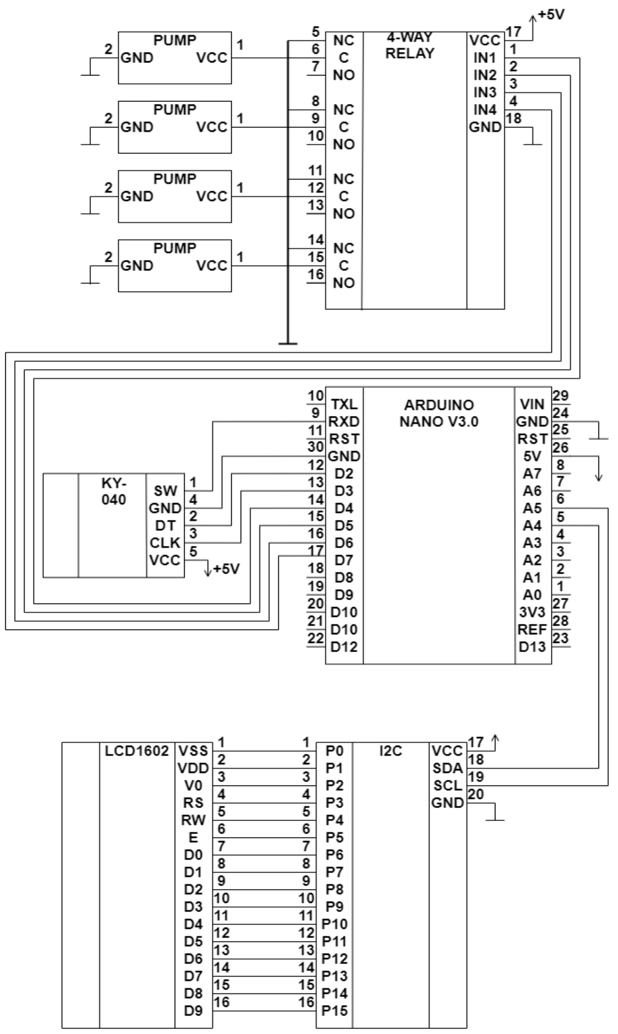


Fig. 10. Schematic diagram of the subsystem

CONCLUSIONS

The main result of research was the development of principles for building a microprocessor subsystem of a smart home to control multi-channel irrigation of houseplants. The subsystem was tested on 16 channels. In addition, both the hardware and software of the system were developed.

The main advantage of this system compared to existing analogs was that watering is multi-channel. Therefore, there was an opportunity to adjust watering separately to each channel after all plants on various are exacting to water. Furthermore, the Arduino NANO microcontroller made the system very compact and convenient. The user had the opportunity to name each channel uniquely for himself, which made the use of the subsystem very comfortable. Interaction with the system was via an LCD screen that was connected to the I2C module. With the help of the created user interface and the encoder module KY040, it was possible to make adjustments to watering and its period.

REFERENCES

[1] M. Makana, N. Nwulu and E. Dogo (2021), “Automated Microcontroller-Based Irrigation System”, Examining the Impact of Deep Learning and IoT on Multi-Industry Applications, 2021, pp. 45-60, doi: 10.4018/978-1-7998-7511-6.ch004. (Accessed: 18 April 2022)

[2] Juniper, A. (2018). The Smart Smart Home Handbook: Control Your Home With Your Voice, 75 p. Available at: https://www.amazon.com/Smart-Home-Handbook-Control-Voice/dp/1781575800. (Accessed: 18 April 2022)

[3] Domb, M. (2019). Smart Home Systems Based on Internet of Things, 89 p., doi: 10.5772/intechopen.84894. (Accessed: 18 April 2022)

[4] Robbins, T. (1997). Faculty of Engineering Science, University of Western Ontario. Automatic Plant Watering System, 35 p. Available at: https://books.google.com.ua/books/about/Automatic_Plant_Watering_System.html. (Accessed: 18 April 2022)

[5] Almy, T. (2020). Far Inside the Arduino: Nano Every Supplement, 10 p. Available at: https://www.amazon.com/Far-Inside-Arduino-Every-Supplement/dp/B08GFL6VBF. (Accessed: 18 April 2022)

[6] Kurniawan, A. (2021). IoT Projects with Arduino Nano 33 BLE Sense, 24 p. available at: https://www.amazon.com/Projects-Arduino-Sense-Step-Step

[7] Blum, J. (2013). “Exploring Arduino: Tools and Techniques for Engineering Wizardry”, CRC Press, 131 p., doi: 10.30609/jeti.v4i04.13174 (Accessed: 18 April 2022)

[8] Boxall, J. (2014). Arduino Workshop: A Hands-On Introduction with 65 Projects 1st Edition, 421 p. Available at: https://www.amazon.com/Arduino-Workshop-Hands-Introduction-Projects/dp/1593274483. (Accessed: 18 April 2022)

[9] K. Chochiang, K. Chaowanawatee, K. Silanon and T. Kliangsuwan (2019). "Arduino Visual Programming", 2019 23rd International Computer Science and Engineering Conference (ICSEC), 2019, pp. 82-86, doi: 10.1109/ICSEC47112.2019.8974710. (Accessed: 18 April 2022)

[10] V. Gurevich (2018). “Electric Relays: Principles and Applications”, CRC Press, 704 p., doi: 10.1201/9781315221168 (Accessed: 18 April 2022).

[11] C. Cameron (2019). "Rotary Encoder", Arduino Applied, 2019, pp. 177-187, doi: 10.1007/978-1-4842-3960-5_9. (Accessed: 18 April 2022).

[12] Y. Badamasi (2014). "The working principle of an Arduino», 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), 2014, pp. 1-4 doi: 10.1109/ICECCO.2014.6997578. (Accessed: 18 April 2022)

[13] N. Pavych, T. Pavych (2019). "Method for Time Minimisation of API Requests Service From Cyber-Physical System to Cloud Database Management System ". Advances in Cyber-Physical Systems, 2019, Volume 4, No 2, pp. 125-131, doi: 10.23939/acps2019.02.125. (Accessed: 18 April 2022)

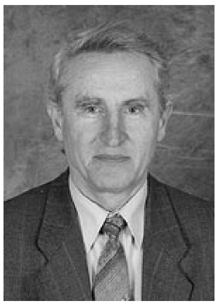
[14] N. Pavych and V. Zahurskii (2021), "Software Architecture for Analyzing the Impact of News on the Stock Market", 2021 11th International Conference on Advanced Computer Information Technologies (ACIT), 2021, pp. 613-617, doi: 10.1109/ACIT52158.2021.9548457. (Accessed: 18 April 2022)

Taras Borak received a Bachelor's degree in Computer Engineering at Lviv Polytechnic National University in 2020. Since 2020 he has been studying to receive a Master's degree in the field of Computer systems and Networks, and has been working as a DevOps Engineer at EPAM Systems. His research interests include embedded subsystems for smart-house and AWS Cloud infrastructure creation.

Dmytro Kushnir. Ph.D. student at Computer Engineering Department of Lviv Polytechnic National University, Institute of Computer Technologies, Automation and Metrology (ICTA). He received a Master's degree in Computer Engineering in 2018 at Lviv Polytechnic National University.

He has been working as a Software Engineer in web development since 2017 at GlobalLogic IT company.

Scientific interests include machine learning, Tracking Algorithms, Ants movement prediction, IoT, mobile and web development.



Yaroslav Paramud Ph.D., Assoc. Prof at Computer Engineering Department of Lviv Polytechnic National University. Scientific interests include processing radar information, research algorithms, and structures of specialized computing devices and systems. He is a Candidate of Technical Sciences, Associate Professor of the Department of Electronic Computers, Institute of Computer Technology, Automation and Metrology, Lviv Polytechnic National University.

Research interests: special information processing, research of algorithms and structures of special computer devices and systems.