

Application of the Nucleo STM32 module in teaching microprocessor techniques in automatic control

Abstract. The paper presents the possibilities of using the Nucleo STM32F746 module as part of the microprocessor systems course during didactic classes in the field of automation and robotics. The components of the laboratory set and the STM32CubeIDE integrated development environment were discussed. The possibility of using the sets to illustrate the tasks like automatic determination of characteristics, FIR and IIR signal filtration and the implementation of the automatic control system has been shown.

Streszczenie. W artykule przedstawiono możliwości użycia modułu Nucleo STM32F746 w ramach przedmiotu systemy mikroprocesorowe podczas zajęć dydaktycznych dla kierunku automatyka i robotyka. Omówiono elementy składowe zestawu laboratoryjnego oraz zintegrowane środowisko programistyczne STM32CubeIDE. Pokazano możliwości wykorzystania zestawów do ilustracji zagadnień automatycznego wyznaczania charakterystyk, filtracji sygnałów FIR i IIR oraz realizacji układu regulacji automatycznej. (Zastosowanie modułu Nucleo STM32 w nauczaniu technik mikroprocesorowych w automatyce)

Keywords: STM32, Nucleo, microcontroller, automatic control system.
Słowa kluczowe: STM32, Nucleo, mikrokontroler, układ regulacji automatycznej.

Introduction

Manufacturers of microcontrollers offer relatively inexpensive experimental modules that facilitate familiarization with the functionality of a given microcontroller and its programming techniques. Currently, one of the popular solutions are the modules of the Arduino family, in which the Arduino UNO with the ATmega328P microcontroller is the most popular [1]. This solution allows to quickly make a prototype of a simple control system, or even a simple digital signal processing (DSP) system [2], but the full use of the microcontroller is quite limited due to the concept of the so-called Arduino bootloader and because of the relatively simple IDE programming environment without the possibility of debugging.

Due to the above limitations, a better solution are experimental modules prepared by the company ST, which has developed a number of modules called Discovery and Nucleo for its microcontrollers. Discovery modules, apart from the microcontroller, contain additional electronic components [3]. An exemplary module is shown in Fig. 1.

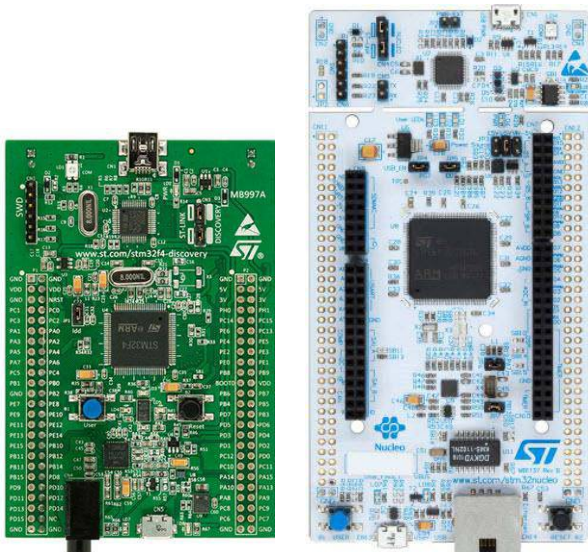


Fig.1. Discovery STM32F429 and Nucleo STM32F746

Nucleo modules, designed in 32, 64 and 144 formats (which is related to the number of I / O pins) typically do not

have any additional electronic components apart from the microcontroller itself, and therefore their price is lower. A characteristic feature of Nucleo is the ST-LINK / V2-1 programmer / debugger, which can be detached (broken off) from the base board. These modules have connectors with derived signals, including connectors of the Arduino UNO standard. For Nucleo modules, PCB covers (so-called shields) can be prepared, which may contain additional I / O elements [4].

Techniques of using and programming STM32 microcontrollers are well presented in numerous tutorials prepared by the ST company and manuals, including Polish ones [5, 6, 7, 8].

Elements of the laboratory kit

For didactic activities, the Nucleo STM32F746 [9] (Fig. 1) module can be used. The microcontroller is based on the ARM® 32-bit Cortex®-M7 CPU core with a floating point unit (FPU), has 1MB Flash memory and 320kB SRAM memory and is equipped with a number of serial interfaces (including I2C, UART, SPI), and, what is important from the point of view of learning about DSP issues, both 12-bit analog-to-digital converters (ADCs) and 12-bit digital-to-analog converters (DACs).

Programming the Nucleo module can be performed using the free STM32CubeIDE environment (Fig. 2), which, using the STM32CubeMX graphic interface, supports the configuration of the pins, individual devices of the microcontroller and clocks (Fig. 3), enabling generation of the initialization code.

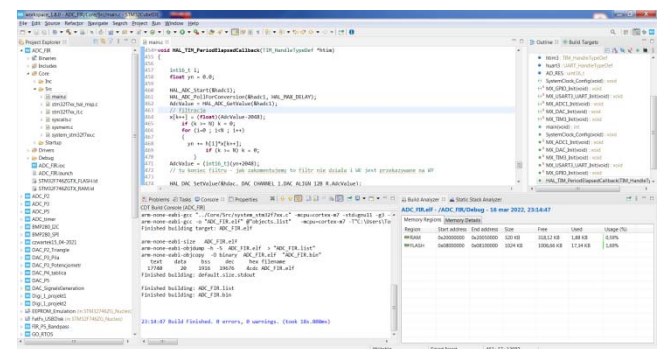


Fig.2. STM32CubeIDE environment

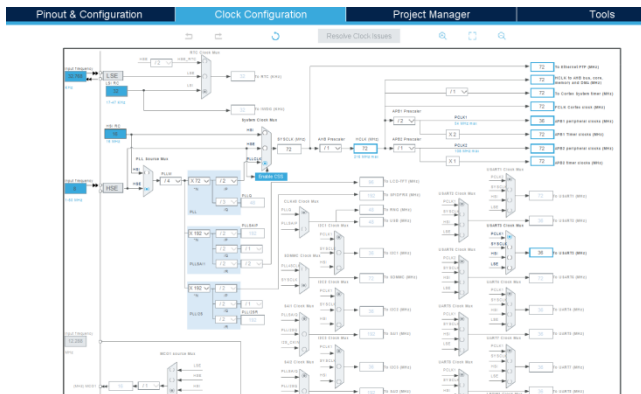


Fig.3. Graphical configuration of clocks in the STM32CubeMX environment

In addition to the Nucleo module, during the laboratory classes, additional modules and electronic components are necessary, for example: the BH1750 light intensity sensor, the BMP280 pressure and temperature sensor, the LM358 and LM386 operational amplifiers, 16×2 LCD displays with an I2C converter, as well as auxiliary elements such as a breadboard, rotary encoder, potentiometers, LED etc.

The first few laboratory classes are devoted to the basic issues of programming techniques in the STM32CubeIDE environment, data acquisition, control (e.g. PWM generation) as well as archiving and visualization of measurements.

An interesting possibility of the STM32CubeIDE environment is the ability to analyze energy consumption using the Power Consumption Calculator (Fig. 4), which allows to accurately assess the energy demand of individual microcontroller devices and shows which of them can be turned off for savings.

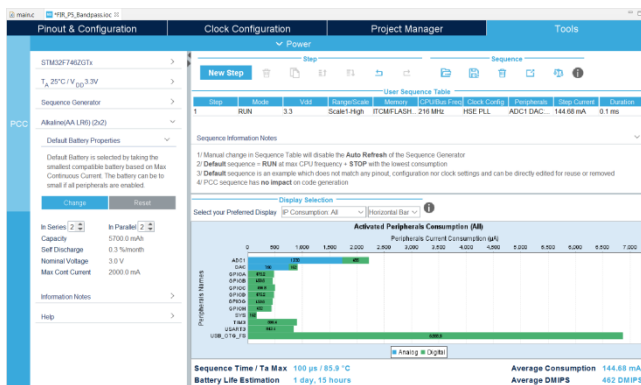


Fig.4. An example of analysis using the Power Consumption Calculator

The ability to use the laboratory set and the environment is important in the following classes related to the implementation of measurement and control systems, digital signal processing algorithms and the preparation of the didactic automatic control system. Examples of laboratory exercises of such systems are presented in the following chapters.

Determination of the photoresistor characteristics

The BH1750 light intensity sensor enables measurements in the range of 1 - 65535 lx, and gives the possibility of a determination of photoresistor sensitivity characteristic. The block diagram of the measurement system is shown in Fig. 5. The reference measurement, carried out by the BH1750, is sent to the microcontroller via the I2C interface. The tested photoresistor, in a divider

circuit with a resistor, e.g. 10 kΩ, is connected to the ADC and should be placed close to the BH1750 module (Fig. 6). The lighting intensity is changed by means of a white LED controlled by a PWM signal. The measurement result, ADC in relation to the lux value, can be observed using Serial Wire Viewer in the STM32CubeIDE environment or sent via UART to other visualization programs.

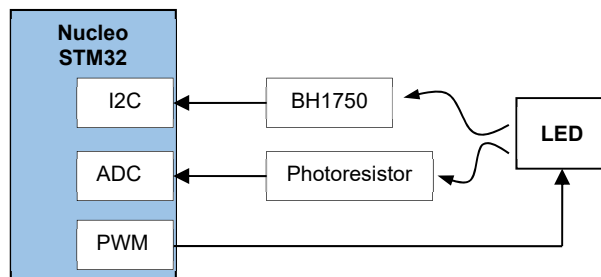


Fig.5. Block diagram of a system for determining the characteristics of a photoresistor

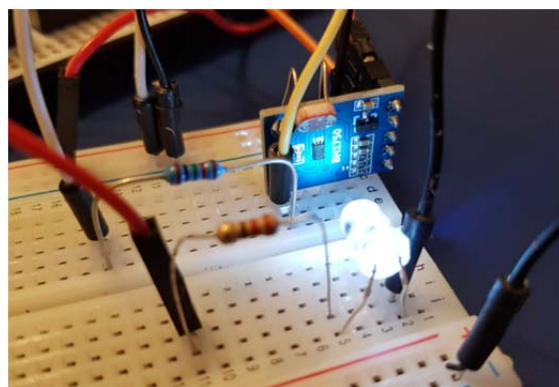


Fig.6. BH1750 module and photoresistor illuminated with white LED

Implementation of digital filtration

Another issue discussed in the class is the presentation of digital signal processing [10, 11]. As mentioned earlier, the STM32F746 microcontroller has analog-to-digital and digital-to-analog converters, which allows to easily build a DSP system, the general concept of which is shown in Fig. 7.

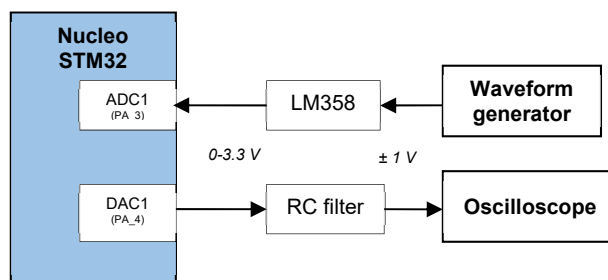


Fig.7. General concept of a digital signal processing system using the Nucleo STM32F746

For the generation of analog test signals, a PC sound card and software for generating sinusoidal signals (for example <https://www.szynalski.com/tone-generator/>) can be used. The voltage range from the sound card ($\pm 1V$) should be adapted to the ADC input voltage range of the analog-to-digital converter (voltage 0-3.3V). The LM358 op-amp can be applied for this purpose. Figure 8 shows the schematic and simulation of the input amplifier operation in the LT Spice environment.

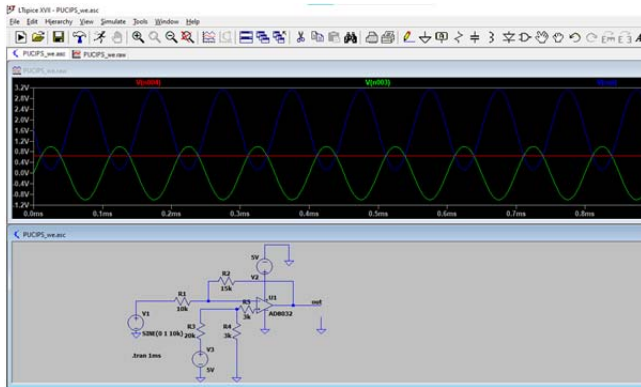


Fig.8. An example of an input amplifier simulation made with the use of LT Spice software

FIR or IIR digital filtration is programmed using the STM32F746 microcontroller. An important element is the correct setting of the ADC and DAC, including the correct sampling rate.

The output signal from the DAC has the character of a step function and a reconstructive filter is needed [12]. Observation of the analog signal can be done with an oscilloscope, and if it is not available, an alternative is computer sound card with oscilloscope visualization software. Such solution has limitations related to the frequency band (typically range up to 20 kHz), but it is sufficient for teaching. An example of such software is shown in Fig. 9, and its additional advantage is the FFT analysis function.

During the course, students implement and analyze the operation of the following filters:

- an averaging filter,
- a band-pass filter,
- IIR filters designed using impulse invariant or bilinear transforms.

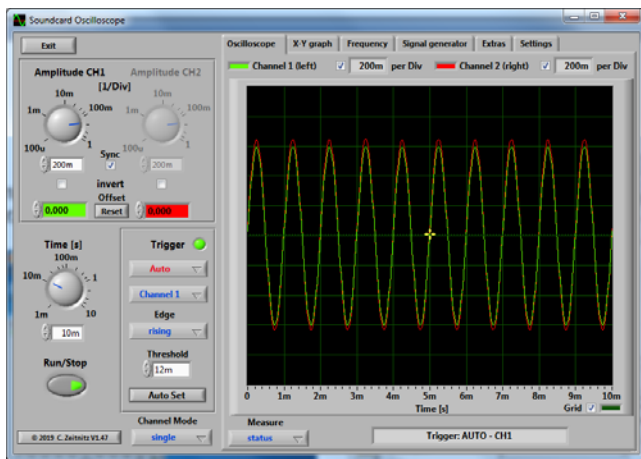


Fig.9. Oscilloscope using a PC sound card (https://www.zeitnitz.eu/Scope_en)

The averaging filter (e.g. of length $N = 5$) is relatively simple to implement and allows to get acquainted with the signal attenuation process. Fig. 10 shows the frequency response of the filter, which can be plotted with the freqz command in Matlab or Octave-Online. A practical assessment of the filter's operation can be made using a generator and an oscilloscope. The input signal is reset for two zeros, for example for 1600 Hz and 3200 Hz, assuming a sampling rate of 8000 samples / second. Additionally, the student can observe the phenomenon of aliasing.

In the next step, the student becomes familiar with the design of FIR filters with given characteristics. The filter

coefficients can be determined by software, for example the FDA Tool from the Matlab package [13]. On the Internet it is also possible to find pages that allow automatic determination of filter coefficients, an example is shown in Fig. 11.

```
octave:> freqz([0.2 0.2 0.2 0.2 0.2],1)
```

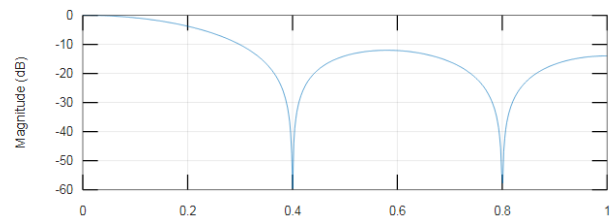


Fig.10. The frequency response of a filter of length $N = 5$ obtained with the freqz command

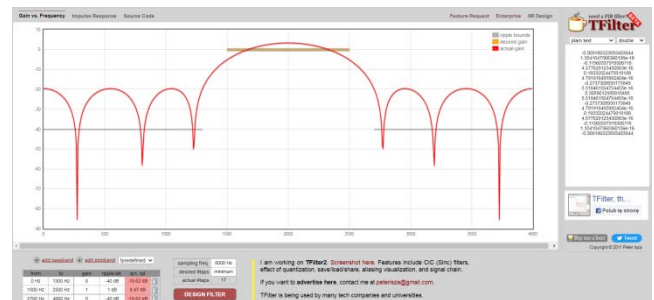


Fig. 11. An example of designing a FIR filter on the page <http://t-filter.engineerjs.com/>

The implementation of IIR filters consists in transforming the transmittance of the analog filter to the form of the transmittance of the digital filter, and then writing the difference equation and its software. The impulse invariant or bilinear transform are used to convert the analog transmittance. Due to the nature of these transformations, the obtained digital transmittances are different and the frequency characteristics are also different [10].

Automatic control system

An interesting exercise, combining elements of the control theory and the practical use of a microprocessor system, is the construction and programming of a demonstration PID temperature control system. The block diagram is shown in Fig. 12.

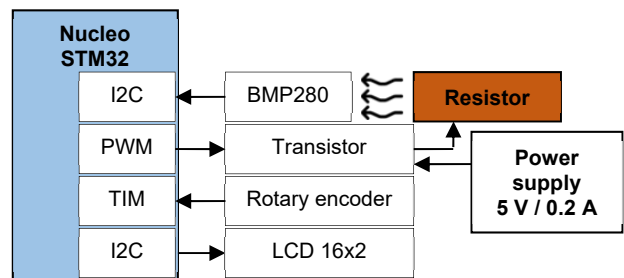


Fig.12. Block diagram of the temperature control system

The student's task is to prepare a hardware and software project that stabilizes the temperature with a specific deviation (e.g. at the level of 5% or less of the control range value). In addition, it is required to set a reference value by means of serial communication or a rotary encoder as well as the ability to observe the current value measured with the STM Studio or Telemetry Viewer software and LED or LCD display. An example of a controller circuit is shown in Fig. 13. For temperature measurement, the DS18B20 or

BMP280 sensors can be applied, which are characterized by a relatively high measurement accuracy.

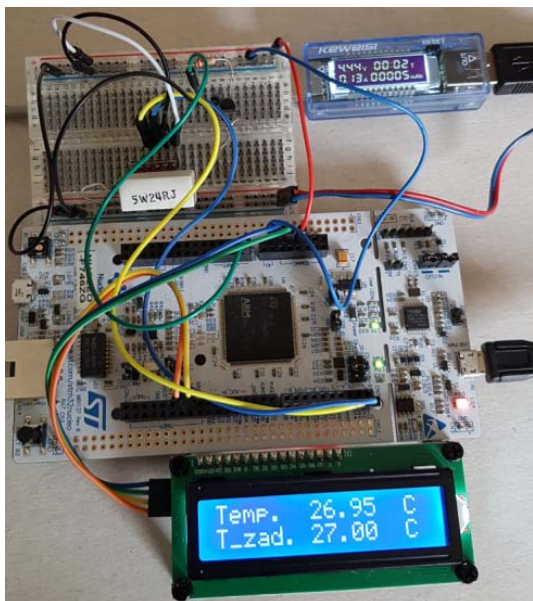


Fig.13. An example of a temperature controller system

Initial analysis of the system operation and parameterization can be performed using the Matlab environment based on the tutorial [14]. The Curve Fitting Tool (Fig. 14) is helpful to determine the parameters of the object from the step response, and the selection of the controller settings is conveniently performed with the PID Tuner tool (Fig. 15).

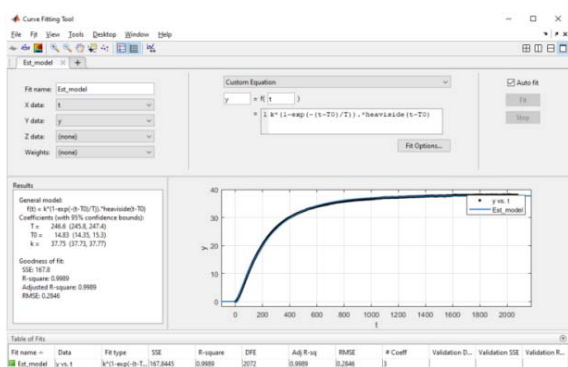


Fig.14. Determination of transmittance parameters using the Curve Fitting Tool

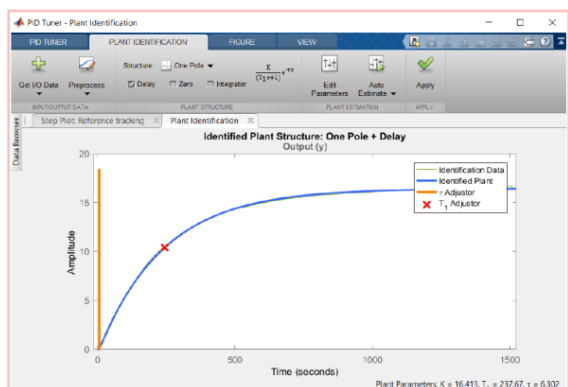


Fig.15. Selection of controller settings by means of PID Tuner

As mentioned before, the stabilized temperature measurement result can be observed using Serial Wire

Viewer in the STM32CubeIDE environment or sent via UART to other visualization programs.

Conclusions

The STM32CubeIDE development environment facilitates the preparation of a project for Nucleo and Discovery prototype modules, and also allows to observe the real-time operation of the designed systems and combine theory with practice. The cost of the devices allows the student to use the proposed solutions also outside laboratory classes at home during a distance learning.

The presented, selected information on the implementation of laboratory classes shows that at a relatively low cost, it is possible to organize an introductory course on signal control processing. The environment should be supplemented with additional programs (both paid and free alternative) enabling quick prototyping and preparation of automatic control parameters.

This research was funded partly by the 2022 subvention and partly with the SMART4ALL EU Horizon 2020 project, Grant Agreement No 872614.

Authors: Tomasz Marciniak PhD, E-mail: tomasz.marciniak@put.poznan.pl, Kacper Podbucki MSc, E-mail: kacper.podbucki@put.poznan.pl, Jakub Suder MSc, E-mail: jakub.suder@put.poznan.pl, Poznan University of Technology, Institute of Automatic Control and Robotics, Division of Signal Processing and Electronic Systems, ul. Jana Pawla II 24, 60-965 Poznań, Poland.

REFERENCES

- [1] Arduino Home, <https://www.arduino.cc/>, access: 30.03.2022
- [2] Marciniak T., Dąbrowski A., "Nauczanie zagadnień cyfrowego przetwarzania sygnałów z zastosowaniem modułów z mikrokontrolerem", *Przegląd Elektrotechniczny*, 94 (2018), nr 9, 125-127
- [3] STM32 Discovery Kits, <https://www.st.com/en/evaluation-tools/stm32-discovery-kits.html#products>, access: 30.03.2022
- [4] Marciniak T., Dąbrowski A., Puchalski R., Dratwiak D., Marciniak W., Zastosowanie mikrokontrolera STM32F410 do prezentacji zagadnień cyfrowego przetwarzania sygnałów, *Przegląd Elektrotechniczny*, 95 (2019), nr 10, 118-120
- [5] Szumski M., *Mikrokontrolery STM32 w systemach sterowania i regulacji*, BTC, (2018)
- [6] Galewski M., *STM32. Aplikacje i ćwiczenia w języku C z biblioteką HAL*, BTC (2019)
- [7] Kurczyk A., *Mikrokontrolery STM32 dla początkujących*, BTC (2019)
- [8] Paprocki K., *Mikrokontrolery STM32 w praktyce*, BTC (2009)
- [9] STM32 Nucleo-144 development board with STM32F746ZG MCU, supports Arduino, ST Zio and morpho connectivity, <https://www.st.com/en/evaluation-tools/nucleo-f746zg.html>, access: 30.03.2022.
- [10] Ünsalan C., Yücel M. E., Gürhan H. D., *Digital Signal Processing using Arm Cortex-M based Microcontrollers Theory and Practice*, Arm Education Media (2018)
- [11] Marciniak T., Podbucki K., Suder J., Dąbrowski A., Analysis of Digital Filtering with the Use of STM32 Family Microcontrollers. In: Bartoszewicz A., Kabziński J., Kacprzyk J. (eds) *Advanced, Contemporary Control. Advances in Intelligent Systems and Computing*, Springer, Cham., (2020), vol 1196, 287-295
- [12] Marciniak T., Cetnarowicz D., Pawłowski P., Dąbrowski A., Projektowanie filtrów analogowych do modułów cyfrowego przetwarzania sygnałów, *Przegląd Elektrotechniczny*, 95 (2019), nr 10, 125-129
- [13] Matlab, <https://www.mathworks.com/products/matlab.html>, access: 30.03.2022
- [14] Burn K, Cox C. A, hands-on approach to teaching system identification using first-order plus dead time modelling of step response data. *The International Journal of Electrical Engineering & Education*, (2020);57(1), 24-40