

Artur ZAWADA¹, Jacek RYSIŃSKI²

Opiekun naukowy: Jacek RYSIŃSKI²

ARDUINO – ROZWÓJ TECHNOLOGICZNY NA PRZESTRZENI DWÓCH DEKAD

Streszczenie: Arduino, płytką rozwojową będącą kamieniem węgielnym pod budowę niezliczonej liczby projektów. W założeniu mikrokontroler ten był narzędziem dydaktycznym. W roku 2023 mija 20 lat od rozpoczęcia prac nad jego produkcją. Publikacja ta ma na celu weryfikację jak potoczył się rozwój platformy pod względem zastosowania w rozwiązaniach docelowych.

Słowa kluczowe: Arduino, mikroprocesor, płytki rozwojowe, AVR

ARDUINO – TECHNOLOGICAL DEVELOPMENT OVER TWO DECADES

Summary: Arduino, the cornerstone development board for building countless projects. In principle, this microcontroller was a teaching tool. In 2023, 20 years will have passed since the start of work on its production. The purpose of this publication is to verify how the development of the platform unfolded in terms of its use in target solutions.

Keywords: Arduino, microprocessor, development boards, AVR

1. Platforma Arduino

Pierwsza oficjalna płytką Arduino została wdrożona do masowej produkcji w roku 2005. Mimo to początki platformy sięgają roku 2003, kiedy to Massimo Banzi, David Cuartielles oraz David Mellis zmodyfikowali platformę rozwojową „Wiring” nazywając ją „Arduino”. Najważniejszą zmianą było dostosowanie zaprojektowanej wcześniej przez Hernando Barragán z „Instytutu Projektowania Interakcji w Ivrei” płytki, o topologię zgodną z architekturą mikrokontrolera Atmel AVR Atmega8. W tym momencie należy również wspomnieć, iż sam mikrokontroler AVR

¹ mgr inż., Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn I Informatyki, artur.zawada88@gmail.com

² dr inż., Akademia Techniczno-Humanistyczna w Bielsku-Białej, Wydział Budowy Maszyn I Informatyki, jrysinski@ad.ath.bielsko.pl

AT90S8515 (oficjalna nazwa) powstał dużo wcześniej, bo w roku 1996. Dopiero stworzenie zintegrowanego środowiska „Processing” w roku 2001 opartego na zasadach polityki „Open source” ułatwiło późniejsze powstanie platformy „Wiring” a następnie „Arduino”. Choć na początku platforma nie wyglądała tak jak dziś, została bardzo ciepło przyjęta przez zwolenników elektroniki, umożliwiając im tworzenie własnych projektów. Najważniejszą zaletą aplikacji „Arduino IDE” powstałej na bazie wcześniejszego środowiska „Processing” było zachowanie polityki „Open Source”, co znacząco poszerzyło grono osób, które chciały spróbować swoich sił przy programowaniu [1-4, 7, 9].

1.1. Budowa płytki

Od samego początku założeniem Arduino było powstanie platformy edukacyjnej wyposażonej w procesor, pamięć operacyjną oraz koniecznych do integracji z otoczeniem portów wejść oraz wyjść. Wiele osób w sposób błędny porównuje ją z komputerami PC. Z tego powodu należy zwrócić uwagę z czego składa się Arduino i do czego zostało zaprojektowane. Pierwszą i najbardziej widoczną różnicą jest budowa. Komputer składa się z mikroprocesora, karty graficznej, pamięci ROM (ang. read only memory), RAM (ang. random access memory), dysku SSD(ang. solid state drive) /HDD(ang. hard disk drive), portami wejść/wyjść. Mikrokontroler natomiast jest urządzeniem, które zawiera wewnątrz jednej obudowy mikroprocesor, pamięć SRAM (ang. static random access memory), pamięć FLASH, porty wejść/wyjść. Moduł nie jest wyposażony również w kartę graficzną, ponieważ służy do całkowicie innych zadań. Często również porównuje się ilość pamięci RAM. W obu urządzeniach zastosowany typ to SRAM (ang. static random access memory). W Arduino znajduje się zaledwie 2KB, natomiast PC posiada od 256KB do 8MB w zależności od poziomu nie licząc zewnętrznej pamięci RAM (kilkadziesiąt GB). Tu właśnie pojawia się jedna z najbardziej znaczących różnic pomiędzy komputerem PC, a mikrokontrolerem. Mianowicie w przypadku komputerów otwarcie programu, czy samego systemu operacyjnego wiąże się z przekopiowaniem go do zewnętrznej pamięci RAM. Każda późniejsza zmiana programu, czy powstanie nowej zmiennej powoduje alokację nowej przestrzeni w pamięci RAM. Minimalizacja programu wiąże się z przeniesieniem zawartości pamięci na dysk twardy (plik stronicowania). Ma to na celu przygotowanie miejsca na inny proces. Sam procesor nie korzysta bezpośrednio z pamięci RAM jednak pobiera z niej wymagane dane do pamięci SRAM zwanej również pamięcią CACHE. W przypadku Arduino program jest uruchamiany bezpośrednio z pamięci FLASH natomiast w pamięci SRAM znajdują się zmienne wykorzystywane w programie. Analogiczną sytuację możemy zauważyć porównując prędkość taktowania procesora Arduino (Uno-16MHz) z komputerami PC(GHz). Mikrokontroler ma za zadanie obsługę jednego procesu (może być wielowątkowy na innych płytkach) natomiast komputer musi w tle zadbać, o obsługę wielu niezależnych programów systemu operacyjnego [1-5].

Z pozostałych elementów jakie wchodzi w budowę omawianego modułu to chip interfejsu USB, regulator napięcia 5V (istnieją płytki pracujące na logice 3.3V) oraz złącze programowania. Złącze to (port szeregowy SPI) może być wykorzystywany do komunikacji z innymi modułami. Standardowa płytka posiada 14 pinów cyfrowych oraz 6 pinów analogowych. Każdy pin może pracować jako wejście lub wyjście (z pewnymi wyjątkami). Niektóre z tych pinów mają jeszcze inne funkcje. Mogą one

pracować w trybie transmisji szeregowej UART, wspomnianego wcześniej protokołu SPI, czy magistrali I²C, a także magistrali OneWire. Do diagnostyki modułu jak również programowania używany jest wbudowanego w port USB protokołu UART do transmisji szeregowej. W sytuacji, gdy zaczyna brakować pamięci programu możliwe jest również usunięcie znajdującego się w pamięci bootloadera obsługującego port USB i wgranie większego programu przez port szeregowy. Zaletą tego rozwiązania jest to, że za moduł pośredniczący we wgraniu programu może posłużyć inny moduł Arduino.

1.2. Rozwój flagowej płytki Arduino UNO

Pierwszym oficjalnym modelem było „Arduino UNO”. Aktualnie moduł ten, mimo iż nie jest najmocniejszą wersją tej platformy, nadal jest uważany za najbardziej znaną wręcz flagową płytkę z tej rodziny. Wynika to z faktu, iż powstało do niej bardzo dużo wszelkiego rodzaju nakładek (ang. Shield) poszerzających jej możliwości. Począwszy od modułów ethernetowych, poprzez sterowniki serw czy panele dotykowe. Sama płytka doczekała się również dwóch aktualizacji, kolejno Rev2 oraz Rev3. Przy czym były to wersje, które wprowadzały znaczące zmiany. W międzyczasie powstawało kilka różnych wariantów, które różniły się między sobą w nieznaczny sposób. W pierwszej kolejności należy wspomnieć o sercu modułu, czyli procesorze. Wspomniany wcześniej procesor Atmel AVR Atmega8 rok po premierze pierwszego modułu doczekał się następcy. Atmel AVR Atmega168 różniła się jedynie większą pojemnością pamięci FLASH (16KB). W roku 2009 powstała alternatywna wersja płytki o nazwie „Arduino Duemilione” wyposażona między innymi w mikroprocesor Atmel AVR Atmega328 posiadający już 32KB pamięci FLASH. Rok później mikroprocesor ten został zastosowany również we flagowym produkcie i pozostaje on w tej formie do dnia dzisiejszego. Wyjątek stanowi wersja z wlutowanym na stałe mikroprocesorem jednak nie różni się ona parametrami. Sam mikrokontroler wykonuje jedną operację matematyczną za każdym razem, kiedy kwarcowy generator drgań poda na jego wejście impuls. Kwarc ten odpowiada zatem za prędkość taktowania całej jednostki, która wynosi 16MHz (tylko platformy z mikroprocesorem AVR Atmega328 i starsze) czyli 16 milionów operacji na sekundę. W znaczący sposób zmienił się sposób komunikacji z modułem. Prototypowa wersja była wyposażona w złącze DB9 i standard komunikacji RS232. Bardzo szybko złącze zostało wyeliminowane na rzecz złącza USB. Pierwotnie zastosowany chipset ATMEGA8U2 USB został wymieniony dopiero w 2010 roku na wersję ATMEGA16U2 jednak umożliwił on znaczne większe możliwości. Powstanie Arduino UNO Rev3 w 2010 roku można stwierdzić jako zakończenie prac nad rozbudową podstawowej wersji, niezawierającej innych funkcjonalności.

1.3. Moduły wejść/wyjść

Pojęcie platformy Arduino nie ogranicza się jedynie do samej płytki, zaliczają się do niej również nakładki, czujniki czy urządzenia odpowiedzialne za ruch. Za ich pomocą można zbudować prawie że dowolny model. Do dziś powstała nieskończona liczba projektów, w których wykorzystano między innymi:

- kontroler Ethernet,
- kontroler Wi-Fi,
- czujniki temperatury,
- host USB,
- wyświetlacz graficzny,
- czujniki ciśnienia,
- moduł Bluetooth,
- wyświetlacze OLED,
- czujniki nacisku,

-czujniki wilgoci,	-czujniki ruchu,	-dalmierze,
-czujniki podczerwieni,	-diody LED,	-przełączniki,
-sterowniki silników,	-przełączniki,	-potencjometry.

Dostępnych w sprzedaży znajduje się również wiele modułów zwiększających liczbę wejść oraz wyjść. Jak zostało wcześniej wspomniane platforma Arduino nie została przewidziana do obróbki grafiki. Mimo to pojawiło się kilka czujników umożliwiających jej przetwarzanie. Jednym z takich czujników wizyjnych jest PIXY który zajmuje się przetwarzaniem grafiki natomiast do Arduino dostarczane są dane mówiące o położeniu obiektu w przestrzeni. Połączenie kamery oraz dwóch serwomechanizmów udowodniło, że niepozorna płytką Arduino ma wystarczającą moc obliczeniową, aby śledzić trajektorię piłeczki podczas meczu tenisa stołowego.

1.4. Arduino IDE

Platforma Arduino to również wspomniane wcześniej środowisko Arduino IDE. Z perspektywy czasu można stwierdzić, że, fenomenem platformy nie była sama płytką, lecz współpracujące z nią środowisko programistyczne. Głównym założeniem projektu miało być jak najprostsza obsługa, ponieważ grupą docelową były osoby młode nieposiadające umiejętności programowania. Platforma miała być przyjazna w obsłudze i zachęcać młodych inżynierów do pogłębiania swojej wiedzy na temat elektroniki. Z tego powodu już w środowisku zaimplementowane zostały różne biblioteki, które są automatycznie dołączane do projektu. Sam kod programu został przygotowany tak, że kompilator tworzy główną funkcję `main()`. Zawiera ona wszystkie potrzebne biblioteki oraz protokoły odpowiadające za komunikację z komputerem. Dla użytkownika przewidziane zostały dwie funkcje. Pierwszą funkcją jest `setup()` i znaleźć się w niej powinny wszystkie funkcje, które mają wykonać się tylko raz, jak na przykład ustawienie prędkości transmisji danych pomiędzy komputerem, a urządzeniem. Funkcja za to odpowiedzialna to `Serial.begin(9600)` gdzie wartość w nawiasie to właśnie prędkość transmisji. Już w tym miejscu możemy zauważyć prostotę tego środowiska. Mianowicie aby rozpocząć komunikację należy podać więcej parametrów, między innymi bity parzystości czy porty, na których ma odbywać się ta wymiana danych. Wszystkie te rzeczy dzieją się automatycznie zmniejszając w ten sposób poziom wymaganej wiedzy użytkownika. Kolejnym przykładem jest zapisany poniżej program, którego celem jest mruganie wbudowaną diodą LED [5]. Przykład nieco trywialny, lecz w łatwy sposób może zademonstrować co projektanci mieli na celu.

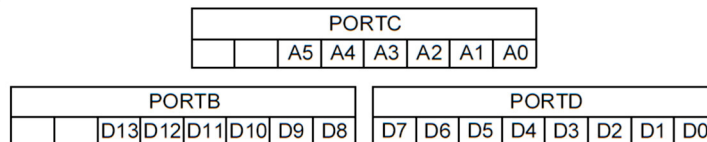
```
1. void setup()
2. {
3.   pinMode(LED_BUILTIN, OUTPUT);
4. }
5. void loop()
6. {
7.   digitalWrite(LED_BUILTIN, HIGH);
8.   delay(1000);
9.   digitalWrite(LED_BUILTIN, LOW);
10.  delay(1000);
11. }
```

Program zaczyna się od wspomnianej wcześniej funkcji `setup()` niezwracającej żadnej wartości o czym informuje przedrostek `void`. Wewnątrz tej funkcji znajduje

się wywołanie kolejnej funkcji mającej na celu przypisanie do pinu LED_BUILTIN (pin 13) pracy w trybie wyjścia. Jak sama nazwa wskazuje jest to wbudowana dioda LED. Od linii 5 rozpoczyna się pętla loop() która jest wykonywana w nieskończoność (podczas pracy urządzenia). Dla wielowątkowych mikrokontrolerów możliwe jest wywołanie funkcji loop1() działającej niezależnie. Wewnątrz tej funkcji znajdują się dwie kolejne funkcje działające naprzemiennie. Pierwszą z nich jest digitalWrite(LED_BUILTIN, HIGH); która ma na celu przypisanie do wyjścia o nazwie LED_BUILTIN wartości wysokiej, czyli logicznej jedynki, a co za tym idzie załączenia diody. Następnie mikrokontroler zatrzymuje pracę na 1000ms, za co odpowiada funkcja delay(1000) której argumentem jest czas wyrażony w ms. W linii 9 dioda zostaje wyłączona po czym w linii 10 praca jednostki znów jest wstrzymana na 1000ms. Po zakończeniu programu rozpocznie się on ponownie od linii 5 włączając ponownie diodę w linii 7. Dla osoby, która nie miała do czynienia z programowaniem w języku C nawet tak krótki i prosty program może nie być zbyt oczywisty. Dlatego dla porównania poniżej przedstawiono program dla tego samego mikrokontrolera, jeżeli wyeliminowalibyśmy jedynie jedną bibliotekę.

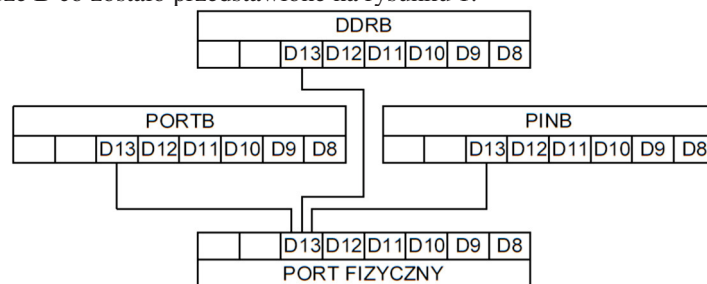
```

1. void setup()
2. {
3.   DDRB=B00100000;
4. }
5. void loop()
6. {
7.   PORTB=B00100000;
8.   delay(1000);
9.   PORTB=B00000000;
10.  delay(1000);
11. }
    
```



Rysunek 1. Porty mikrokontrolera Arduino

Biblioteka, która została usunięta odpowiedzialna była za sterowanie w przyjazny i intuicyjny sposób stanem poszczególnych wyjść. W przypadku braku takiej biblioteki konieczna jest praca na rejestrach mikrokontrolera. Pin 13 znajduje się w rejestrze B co zostało przedstawione na rysunku 1.



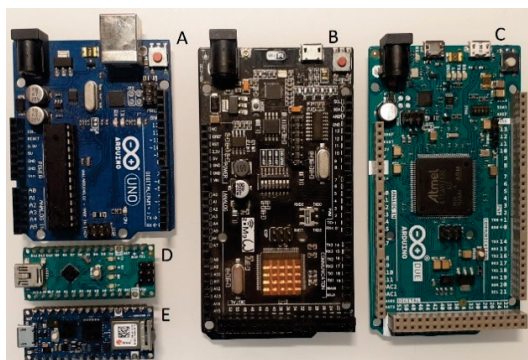
Rysunek 2. Rejestry portów fizycznych

Na rysunku 2 natomiast przedstawione zostało to, że każdemu pinowi odpowiadają trzy rejestry i nazywają się one kolejno DDR, PORT oraz PIN. Rejestr DDR decyduje o trybie pracy danego pinu. Jeżeli na odpowiadającym danemu pinowi bicie znajduje się jedynka pracuje on w trybie pracy wyjścia. Stąd w funkcji `setup()` widnieje polecenie `DDRB=B00100000;` gdzie na bicie szóstym od prawej strony pojawiła się jedynka. Port PIN to rejestr, który przechowuje stan pinu, jeżeli pracuje on jako wejście, dlatego nie został on użyty w powyższym programie. Natomiast rejestr o nazwie PORT określa stan pinów pracujących jako wyjście stąd wartość szóstego od lewej bitu przyjmuje na przemienne wartości 0 oraz 1 mrugając przez to podpiętą do wyjścia diodą.

W przypadku tak krótkiego programu posługiwanie się adresami nie jest trudne. Problem pojawia się, kiedy używanych portów jest wiele i łatwo można się pomylić. Najważniejszą zaletą jednak jest fakt, iż programista nie musi już znać języka niskiego poziomu. Większość, a praktycznie całość programu można napisać za pomocą języka wyższego poziomu. Na tej podstawie można postawić tezę, że „środowisko programistyczne było największym fenomenem tej platformy”.

2. Rozwój platformy

W poprzednim rozdziale opisana została flagowa płytki producenta (Rysunek 3A). Oczywiście jest, że nie jest to jedyny produkt z rodziny Arduino. W 2008 roku zadebiutowała płytki Nano (Rysunek 3D) charakteryzująca się przeszło czterokrotnie mniejszymi rozmiarami (UNO 68,6x53,4mm, NANO 43,2x18,5mm) [6]. Parametrami hardwarowymi obie płytki nie różnią się znacząco. Nieznacznie więcej pamięci w UNO. Do dyspozycji płytki NANO są natomiast dwa dodatkowe piny analogowe. Znajdujący się na rysunku 3B mikrokontroler jest klonem Arduino MEGA (101,6x53,3mm). Modułu, który powstał w 2010 roku jako większy odpowiednik UNO. Rozstaw złączy, mocowań, a także napięcie pracy pozostało bez zmian. Takie rozwiązanie sprawiło, że większość nakładek zaprojektowanych dla UNO, może z powodzeniem zostać zastosowana w MEGA. Znacząco natomiast zmienił się mikroprocesor o nazwie Atmel AVR Atmega2650, który nie tylko jest już w stanie obsłużyć 70 pinów (w tym 16 analogowych), ale także posiada ośmiokrotnie więcej pamięci FLASH(265KB) oraz czterokrotnie więcej pamięci RAM (8KB). Jako że nadal jest to mikroprocesor AVR zastosowany kwarc(16MHz) pozostał bez zmian [1-4, 7, 9].

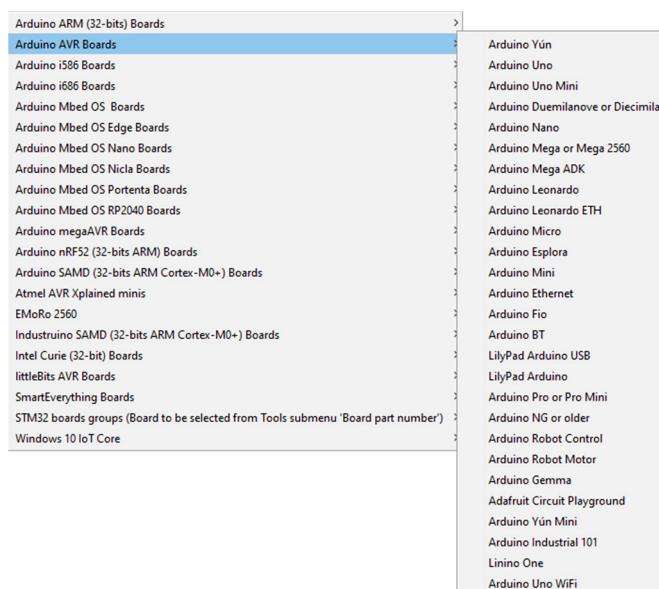


Rysunek 3. Przykładowe mikrokontrolery Arduino

Polityka „Open source” pozwala innym producentom na wytwarzanie swoich odpowiedników oryginalnych płytek Arduino. Jednak każda płytkę musi spełniać kilka wymogów dotyczących jej parametrów. Na tej zasadzie powstał właśnie klon znajdujący się na rysunku 3B. Jest on zgodny z polityką Arduino jednak nie może tak zostać nazwany. Opisywaną jednostkę wyprodukowała firma WeMos została dodatkowo wyposażona w moduł bezprzewodowej komunikacji Wi-Fi.

2.1. Zastosowane mikroprocesory

W roku 2012 marka wypuściła pierwszą płytkę opartą na innym mikroprocesorze którą nazwano Arduino DUE (Rysunek 3C). Do jej wykonania zastosowano płytkę o rozmiarach wcześniej wspomnianego modułu MEGA jednak poza rozłożeniem pinów mikrokontroler ten całkowicie różni się od swoich poprzedników. Pierwszą wspomnianą już różnicą jest 32 bitowy Atmel ARM ATSAM3X8E (Cortex-M3) który zastąpił wcześniejszego AVR. Nowy układ posiada już 512KB pamięci FLASH oraz 96KB RAM. Zmiana jednostki pozwoliła na zwiększenie częstotliwości pracy do 84MHz co znacząco przyspieszyło prędkość wykonywania obliczeń. Niestety mimo zastosowanego rozstawu pinów z MEGA nie jest już możliwe zastosowanie nakładek. Należy również zwrócić uwagę na podłączane do płytki czujniki, gdyż nowa jednostka posiada napięcie pracy równe 3.3VDC. Nie posiada ona również pamięci EEPROM (ang. electrically erasable programmable read-only memory). Niektóre z zaprojektowanych wcześniej bibliotek przestają być kompatybilne, co sprawia duże problemy dla młodych programistów. Nie mniej jednak taki zabieg był konieczny, ponieważ w 2009 roku powstała platforma Raspbery PI. Chociaż nie jest to już mikrokontroler, a mały przenośny mini komputer to dla wielu użytkowników końcowych daje on takie same, a nawet większe możliwości. Nie wliczając modeli prototypowych które fani elektroniki wykonali na własne potrzeby.



Rysunek 4. Rodzaje zastosowanych mikroprocesorów

Na dzień dzisiejszy szacuje się, że powstało przeszło 300 różnego rodzaju płytek (w tym tylko kilkanaście oficjalnych). Wszystkie z nich (a nawet konkurencyjne) można zaprogramować środowisku Arduino IDE. Na rysunku 4 przedstawione zostały wszystkie oficjalne dostępne do pobrania mikroprocesory. Należy również wspomnieć, że istnieje wiele nieoficjalnych płytek, które również można doinstalować do środowiska (druga od dołu STM32). Po prawej stronie rysunku 4 można zauważyć 27 różnego rodzaju płytek, które powstały w oparciu o mikroprocesory z rodziny AVR. Niestety większość z nich nie cieszy się popularnością. Dla przykładu Arduino Ethernet jest modułem UNO wyposażonym w moduł komunikacyjny. Jednak w jego przypadku dużo taniej można zastosować popularny klon Arduino UNO wraz z nakładką Ethernetową.

2.2. Rodzina Arduino

Przeglądając rodzinę płytek Arduino można wprowadzić pewien podział ze względu na zastosowanie, co zostało przedstawione w tabeli 1. Pierwszą gałęzią będą płytki prototypowe które charakteryzuje modułowość, łatwość rozbudowy, a przede wszystkim społeczność, która ciągle je rozwija. Drugą gałęzią jest edukacja. Chociaż od samego początku jest to platforma edukacyjna, stworzono płytki, które są bardziej intuicyjne, bezpieczniejsze oraz nowocześniejsze. Kolejnym rynkiem rozwoju platformy jest przemysł. Modele takie jak Arduino Potenta zostały przewidziane do współpracy z innymi urządzeniami, niektóre z nich umożliwiają sterowanie wyjściami 24VDC inne z kolei posiadają klasę szczelności IP68. Jednostki te wyróżniają się również szybkością taktowania procesora (Potenta x8, 4x ARM Cortex-A53 – do 1,8GHz + ARM Cortex-M4 – do 400MHz). IoT (ang. Internet of Things) również nie został pominięty przez platformę. Z tego powodu powstały modele płytek, które przystosowane zostały do protokołów internetowych. Pobierają one znacznie mniej energii, a niektóre z nich pozwalają na zasilanie bezpośrednio z kabla ethernetowego - technologia PoE (ang. Power over Ethernet). Platforma wprowadziła na rynek również specjalną serię o nazwie LilyPad która charakteryzuje się małymi rozmiarami i niskim poborem energii. Tego typu płytki często wszywane są w ubrania pełniąc rolę sterownika efektów świetlnych.

Tabela 1. Podział Arduino

Prototypowe	Rodzina modułów Arduino			
	Edukacyjne	Przemysłowe	IoT	Do noszenia
UNO, Leonardo, Mini/Pro Mini, NANO, MICRO, MEGA2560, MEGA ADK, Primo, DUE, Seria BLE, RP2040.	Esplora, Robot, Seria BLE, RP2040.	Yun/Yun Mini, Zero, M0/M0 Pro, Tian, Industrial 101, Potenta.	MKR1000, MKRZero, MKRFOX1200, UNO Wi-Fi, Ethernet, Primo.	LilyPad, LilyPad Simple, LilyPad Snap, LilyPad USB.

2.3. Porównanie wydajności

Wprowadzony w tabeli 1 podział wyniku pośrednio z przeznaczenia danego modułu. Bezpośrednio natomiast powiązany jest on z parametrami, wyposażeniem oraz mikroprocesorem (rysunek 4). Stąd też można zauważyć, że porównanie modułów pod względem wydajności jest możliwe jedynie wewnątrz danej rodziny. Na tej podstawie wykonany został test wydajności mikrokontrolerów pierwszej i zarazem najbardziej znanej gałęzi jaką są moduły prototypowe. Do porównania wybrane zostały następujące podzespoły:

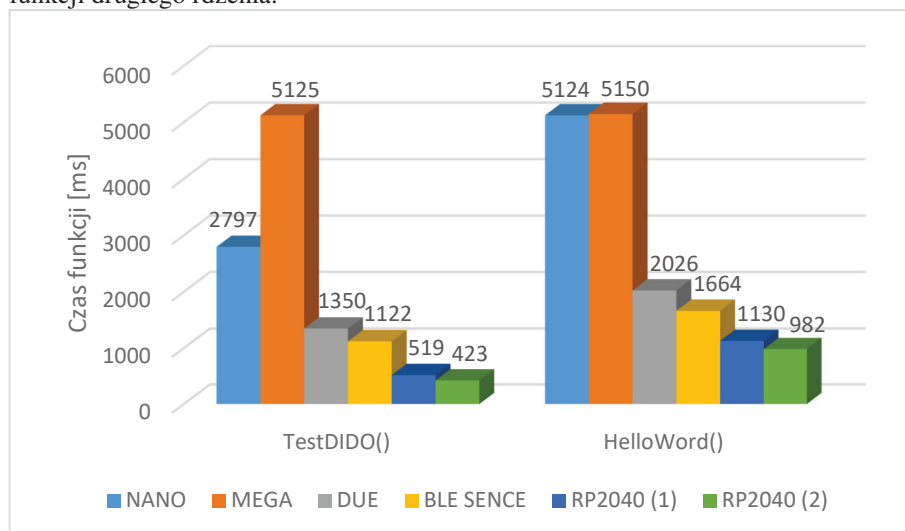
- ARDUINO NANO (AVR Atmega328P),
- ARDUINO DUE (ARM ATSAM3X8E),
- ARDUINO RP2040 Connect (Raspberry Pi RP2040).

Na potrzeby testu wykonano dwie poniższe funkcje. Celem pierwszej z nich była zmiana stanu wyjścia na przeciwny. Celem drugiej było przepisanie do zmiennej typu String zwrotu „Hello World”. W obu przypadkach polecenie zostało wykonane pół miliona razy.

```
1. long TestDIDO()
2. {
3.   long i=0, start;
4.   start=millis();
5.   while(i<=500000)
6.   {
7.     digitalWrite(LED_BUILTIN, !digitalRead(LED_BUILTIN));
8.     i++;
9.   }
10.  return=millis()-start;
11. }
12. long HelloWorld()
13. {
14.   long i=0, start;
15.   String value;
16.   start=millis();
17.   while(i<=500000)
18.   {
19.     value="Hello World";
20.     i++;
21.   }
22.   return=millis()-start;
23. }
```

Przedstawione na rysunku 5 rezultaty są uśrednionym wynikiem trzydziestu cykli każdego mikrokontrolera. Wartości te potwierdzają znaczną przewagę nowszych modułów nad serią AVR. W obu przypadkach zanotowano przeszło dwukrotną przewagę. Oczywistym jest że wynika ona głównie z częstotliwości taktowania procesora jednak nie jest to liniowa zależność. Na rysunku 5 można zauważyć dwie interesujące zależności. Pierwszą z nich jest znacznie dłuższy czas funkcji TestDIDO dla płytki Arduino MEGA. Drugą znacznie ciekawszą zależnością są jednak rezultaty płytki RP2040. Jak można zauważyć na Rysunku moduł ten pojawia się dwa razy. W obu przypadkach badany dokładnie ten sam moduł. Pierwszy z nich oznaczony numerem 1 to płytka wybrana z menadżera płytek Arduino IDE z grupy MBED OS która jest oficjalną biblioteką dla tego modułu. Druga natomiast to nieoficjalna biblioteka przygotowana przez producenta procesora RP2040. Uzyskana rozbieżność

jest efektem wspomnianych wcześniej uproszczeń środowiska. Warto również zauważyć iż procesor ten posiada wbudowane dwa rdzenie posiadające jednakowy dostęp do wszystkich elementów mikrokontrolera (czas na wykresie dotyczy tylko pierwszego rdzenia). Biblioteka przygotowana przez producenta procesora umożliwia start drugiego rdzenia, a czas jego operacji był nieznacznie krótszy od rdzenia pierwszego. W przypadku oficjalnej biblioteki Arduino kompilator nie rozpoznaje funkcji drugiego rdzenia.



Rysunek 5. Wyniki przeprowadzonych testów

2.4. Porównanie Arduino z płytkami deweloperskimi innych producentów

Platforma Arduino była jedną z pierwszych ogólnie dostępnych platform pozwalającą na programowanie mikrokontrolerów. Nic więc dziwnego, że pojawiły się konkurencyjne platformy nowocześniejsze od pierwowzoru. Jedną z nich jest wspomniana wcześniej platforma Raspberry Pi. Inne to między innymi ESP32 czy STM32. Porównanie dwóch modeli Arduino można uznać za nieszczególnie trudne zadanie. Nic bardziej mylnego. Do dziś powstało wiele modeli które posiadają często ten sam rozmiar, oraz podobny mikroprocesor. Różnią się często dodatkami w postaci modułu Wi-Fi, Bluetooth, akcelerometru, żyroskopu itp. W rzeczywistości decydując się na zakup jednostki często wyznacznikiem są dodatkowe wyposażenie oraz cena. Analogiczna sytuacja dotyczy modułów innych producentów. Dla przykładu w tabeli 2 przedstawione zostało porównanie czterech modułów, w tym dwóch z rodziny Arduino oraz dwóch z STMicroelectronics. Wyznacznikiem był rozmiar zgodny z Arduino NANO (na podstawie sklepu internetowego Botland – www.botland.com.pl). Pierwszą najbardziej znaczącą różnicą jest cena (dane z dnia 01.10.2022). Modele z platformy Arduino są znacznie droższe od konkurencji. Z tego powodu na rynku znacznie częściej sprzedawane są klony NANO w cenie około 40PLN. Modele z platformy STM32, mimo iż są tańsze to również znacznie wydajniejsze. Wyjątkiem jest moduł RP2040 (Rysunek 3E) powstały w roku 2021. Jest on połączeniem wydajnego procesora fundacji Raspberry Pi oraz powstałych 2 lata wcześniej modułów serii NANO BLE. Jak widać został on wyposażony w

znacznie mocniejszą jednostkę niż konkurencja. Posiada on również dodatkowe wyposażenie. Między innymi 6 – osiowy czujnik IMU który może posłużyć do uczenia maszynowego. Warto zauważyć, że są to przykładowe mikrokontrolery i istnieją również znacznie mocniejsze moduły platformy STM. Mają one jednak większy rozmiar często zgodny z UNO lub MEGA. Marka Arduino w ostatnich latach zainwestowała znacznie w rozwój. Na tej podstawie powstał między innymi kurs wraz z dedykowanym zestawem do uczenia maszynowego. Kurs ten powstał we współpracy z google oraz uniwersytetem Harvardu. Przedstawione są w nim między innymi zagadnienia TinyML (ang. Tiny Machine Learning). Na tej podstawie kursant może zastosować mikrokontroler z grupy BLE lub RP2040 do rozpoznawania twarzy, gestów czy ruchu. Kilka dni przed publikacją poniższego artykułu miała miejsce premiera pierwszego sterownika PLC marki Arduino, który powstał we współpracy z marką Finder o nazwie OPTA jednak jego szczegóły nie są jeszcze znane.

Tabela 2. Porównanie mikrokontrolerów dwóch producentów

Parametr	Model			
	Arduino		STM32	
	NANO	RP2040	L031K6	F303K8
Mikroprocesor	Atmega 328P	Raspberry Pi RP2040	ARM Cortex M0+	ARM Cortex M4
Częstotliwość pracy	16MHz	133MHz	32MHz	72MHz
Pamięć SRAM	2KB	264KB	8KB	12KB
Pamięć FLASH	32KB	2MB	32KB	64KB
Pamięć EEPROM	1KB	-	-	-
Wyprowadzenia cyfrowe	22			
Wyjścia PWM	6	20	13	17
Wejścia analogowe	8			
Komunikacja szeregową	I ² C, SPI, UART	I ² C, I ² S, SPI, UART	I ² C, SPI, UART, USART	I ² C, SPI, 3xUART, CAN
Zasilanie VIN	6 – 20 VDC	4,5 – 21 VDC	1,65 - 3,6 VDC	2.0 - 3.6 VDC
Napięcie pracy	5 VDC	3.3 VDC		
Dodatkowe wyposażenie		6-osiowy czujnik IMU, układ U-Blox Nina, Wi-Fi, Bluetooth, mikrofon.		
Cena	129,00 PLN	179,00 PLN	83,90 PLN	74,90 PLN

3. Podsumowanie

Arduino – platforma często bagatelizowana przez konkurencję. Sukces stał się nie jako kłątwa, ponieważ do dziś jest ona uważana za edukacyjne płytki prototypowe. Często przedstawiciele konkurencji uchylają się chociażby od porównania platform między sobą, czego autor poniższej publikacji był świadkiem. Ciężko jest się

sprzeciwić takiemu stwierdzeniu porównując produkt, który swoje początki ma prawie dwie dekady temu. Biorąc pod uwagę tempo rozwoju technologicznego zadziwiający jest, iż, platforma do niedawna nie zmieniała znacząco w swoich produktach. Fakt, iż płytki z procesorem powstałym przeszło dwanaście lat temu nadal są produkowane znaczy, że był to dopracowany wyrób. W swojej historii platforma napotkała trudności (tymczasowy rozłam w zarządzie odnośnie praw do produkcji), znacznie wydajniejsze modele konkurencji. Mimo to można stwierdzić, że Arduino którego polityką jest licencja typu „open source” ma się dobrze. Główną zaletą tego jest przyjazne dla nowych użytkowników środowisko. Do dziś powstała niezliczona liczba projektów. Bardzo popularne jest również stwierdzenie „Jeżeli chcesz coś zrobić na Arduino, to możesz być pewien, że ktoś to już wcześniej zrobił”. Ciężko jest się nie zgodzić z tym stwierdzeniem. Mnogość wszelkiego rodzaju bibliotek, przykładów, filmów szkoleniowych sprawia, że większość napotkanych trudności bardzo szybko i sprawnie można rozwiązać dzięki pomocy osób, które wcześniej napotkały podobny problem. Otwartość na inne płytki sprawia, że środowisko nadal się poszerza wchłaniając stosowane w nich biblioteki. Często osoby, które zaczęły swoją przygodę od prostego Arduino pozostają mu wierni. Oczywiście jest wiele osób, które mają prawo twierdzić, że konkurencja ma lepsze rozwiązania, a prostota platformy ogranicza jej możliwości. Nie zmienia to jednak faktu, iż nie wpływa znacząco to na liczbę zwolenników prawdopodobnie najpopularniejszej platformy jaką jest Arduino. Biorąc również pod uwagę zainteresowanie kursem TinyML grupa ta w niedługim czasie może znacznie wzrosnąć.

LITERATURA

1. MONK S.: Arduino dla początkujących – Podstawy i szkice. Helion, Gliwice 2014.
2. MONK S.: Arduino dla początkujących – Kolejny krok. Helion, Gliwice 2015.
3. SALA F., SALA-TEFELSKA M.: Wprowadzenie do mikrokontrolerów AVR. Helion, Gliwice 2021.
4. EVANS M., NOBLE J., HOCHENBAUM J.: Arduino w akcji. Helion, Gliwice 2014.
5. Serwis internetowy Arduino: <https://www.arduino.cc/en/hardware>, 30.09.2022.
6. Serwis internetowy Botland:
<https://botland.com.pl/arduino-seria-nano-oryginalne-plytki/19487-arduino-nano-rp2040-connect-abx00052-7630049203051.html>, 01.10.2022
7. Serwis internetowy Raspbery Pi vs Arduino:
<https://raspberrypivsarduino.com/historia-arduino/>, 01.10.2022.
8. Serwis internetowy Starting electronics:
<https://startingelectronics.org/articles/arduino/uno-r3-r2-differences/>, 03.10.2022.
9. Serwis internetowy Core- electronics: <https://core-electronics.com.au/guides/history-of-arduino/>, 03.10.2022