



ZASTOSOWANIE PLATFORM CYFROWYCH ARDUINO I RASPBERRY PI W NAUCZANIU STEROWANIA OBIEKTEM PNEUMATYCZNYM

**Adam MUC, Lech MURAWSKI, Grzegorz GESELLA,
Adam SZELEZIŃSKI, Arkadiusz SZARMACH**



Wykorzystanie popularnych środków komunikacji i pracy zdalnej w celu udostępnienia, na potrzeby edukacyjne, mechanicznych instalacji laboratoryjnych.

Przegląd platform cyfrowych i rozpoznanie możliwości ich programowania za pośrednictwem Internetu.



Popularność Arduino i Raspberry Pi spowodowała, że pojawiły się rozwiązania konkurencyjne:

- STM32,
- AVR (Adafruit GEMMA i Trinket),
- Banana Pi,
- BeagleBoard,
- RloTboard,
- Stellaris LaunchPad,
- Freedom Freescale,
- PIC
- czy platformy firmy Intel, takie jak Galileo i Edison.



Model	Uno	Leonardo	Mega
Mikrokontroler	ATmega328	ATmega32u4	ATmega2560
Pamięć SRAM	2 kB	2,5 kB	8 kB
Pamięć FLASH	32 kB	32 kB	256 kB
Pamięć EEPROM	1 kB	1 kB	4 kB
Porty I/O	14	20	54
Wyjścia PWM	6	7	15
Wejścia AC	6	12	16
Interfejs szeregowy	UART, SPI, I2C	UART, SPI, I2C	4xUART, SPI, I2C





Najważniejsze cechy:

- wykorzystanie programu rozruchowego (*bootloadera*) zamiast programatora
- darmowe środowisko programistyczne Arduino IDE
- język programowania jest połączeniem środowiska Wiring i języka C oraz wykorzystuje wysokopoziomową bibliotekę
- struktura programu oparta jest na szablonie wykorzystującym funkcje `setup()` i `loop()`





Model	Raspberry Pi 2 A+	Raspberry Pi 2 B+
Mikrokontroler	Broadcom CoS BCM2835	ARM Cortex-A7
Częstotliwość zegara	700 MHz	900 MHz
Pamięć SDRAM	512 MB	1 GB
Wyjście wideo	RCA, HDMI	
Wyjście dźwięku	jack 3,6 mm, HDMI	
Porty USB	1	4
Nośnik danych	microSD	
Złącze sieciowe	brak	10/100 Ethernet (RJ45)
GPIO	40	40
Pozostałe złącza (w ramach GPIO)	UART, I2C, SPI, +3,3V, +5V	
Zasilanie	microUSB	microUSB





Najważniejsze cechy:

- posiada system operacyjny i programowalne porty wejścia/wyjścia
- system operacyjny umiejscowiony jest na karcie SD lub microSD
- duży wybór systemów operacyjnych: Raspbian, Debian GNU, Arch Linux ARM, FreeBSD, Gentoo Linux, Google Chrom OS, Slackware ARM
- podstawowym językiem programowania jest Python





W elektropneumatycznych układach sterownia **częścią energetyczną jest część pneumatyczna, a częścią sterującą jest część elektryczna.**

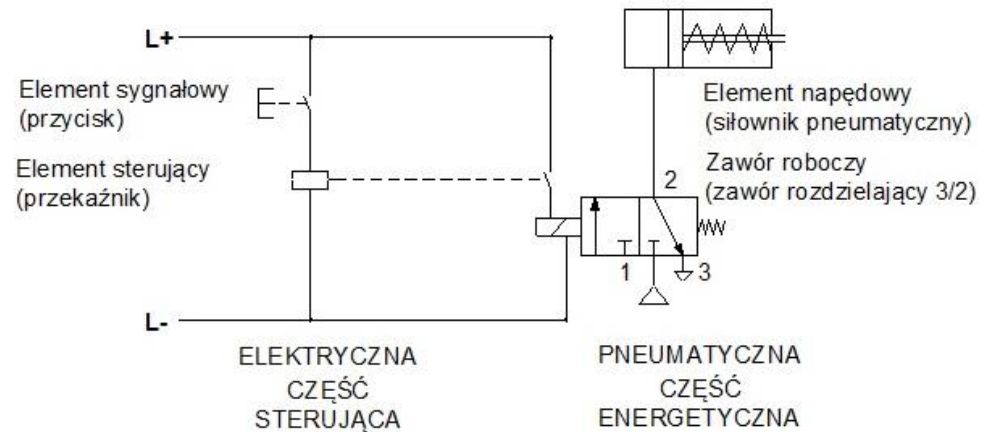
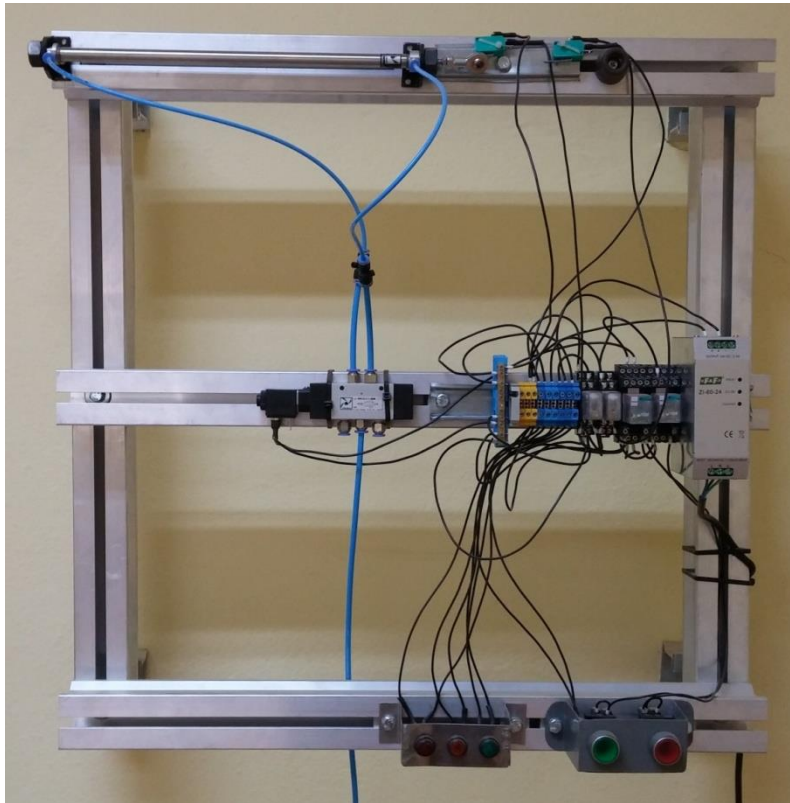
Podstawową techniką sterowania elektrycznej części sterującej jest **technika stykowo-przełącznikowa.**

Niedogodnością klasycznego, stycznikowo-przełącznikowego sterowania elektropneumatycznego jest pracochłonna modyfikacja układu.





STANOWISKO DYDAKTYCZNE



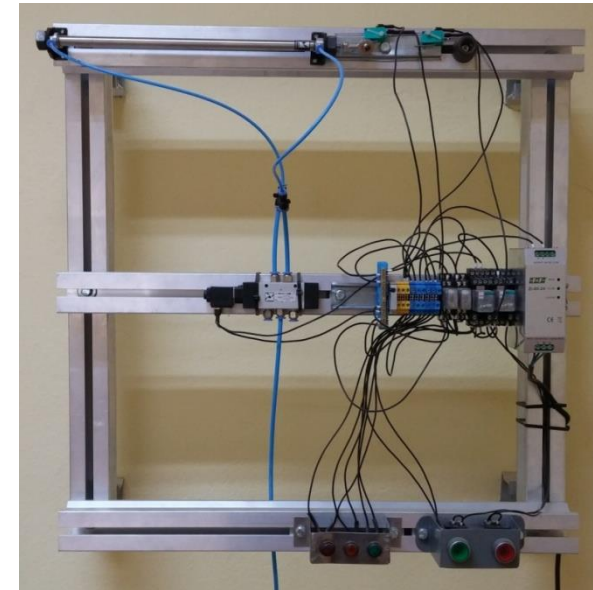


1. Budowa systemu pneumatycznego

- zapoznanie z elementami pneumatycznymi
- układem połączeń i zasadą działania
- montaż systemu pneumatycznego

2. Implementacja manualnego sterowania systemem pneumatycznym za pomocą przekaźników

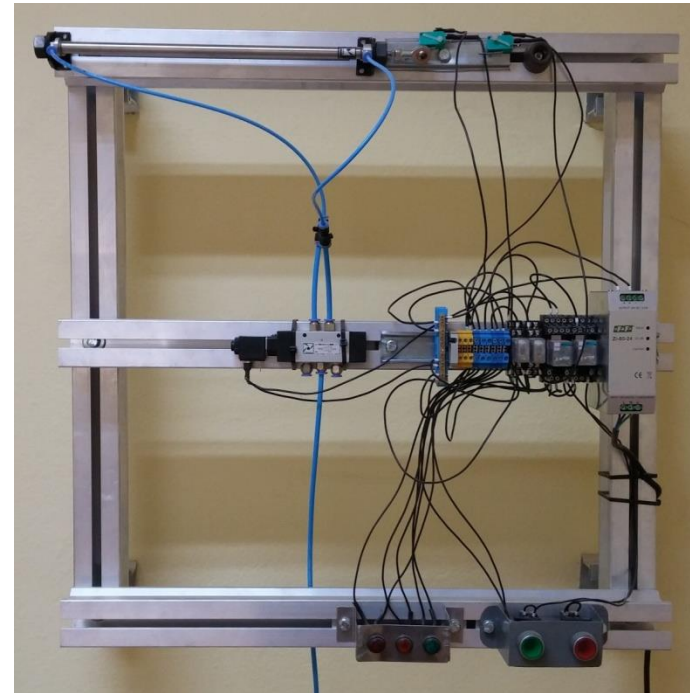
- analiza możliwości sterowania systemem pneumatycznym (szeregowanie zdarzeń)
- implementacja stykowo-przekaźnikowego sterowania układem
- zapoznanie z rodzajami i zasadami stosowania zaworów i czujników w sterowaniu





3. Implementacja programowego sterowania systemem pneumatycznym z poziomu Arduino lub Raspberry Pi

- zapoznanie z możliwościami platform cyfrowych
- podłączenie platformy do systemu
- projekt algorytmu sterowania i programu
- oprogramowanie platformy





PRZYKŁADOWY EFEKT PRACY STUDENTÓW

```
void setup() {
  pinMode (13, OUTPUT);//zawor silownika lewo
  pinMode (12, OUTPUT);//zawor silownika prawo
  pinMode (11, OUTPUT);//dioda zielona
  pinMode (10, OUTPUT);//dioda pomaranczowa
  pinMode (9, OUTPUT);//dioda czerwona
  pinMode (8, INPUT);//przycisk 1
  pinMode (7, INPUT);//przyciks 2
  pinMode (6, INPUT);//krancowka lewo
  pinMode (5, INPUT);//krancowka prawo
}

void loop() {
  if(digitalRead(8)==HIGH)
  {
    pneumatyka();
    krancowki();
  }
  if(digitalRead(7)==HIGH&&digitalRead(8)==LOW)
    animacja();
}

void animacja()
{
  digitalWrite(11,HIGH);
  delay(500);
  digitalWrite(10,HIGH);
  delay(500);
  digitalWrite(9,HIGH);
  digitalWrite(11,LOW);
}
```

```
void pneumatyka() {
  digitalWrite(12,HIGH);
  delay(2000);
  digitalWrite(12,LOW);
  delay(1000);
  digitalWrite(13,HIGH);
  delay(2000);
  digitalWrite(13,LOW);
  delay(1000);
}

void krancowki() {
  if(digitalRead(6)==LOW&&digitalRead(5)==LOW)
    digitalWrite(10,HIGH);
  else
    digitalWrite(10,LOW);

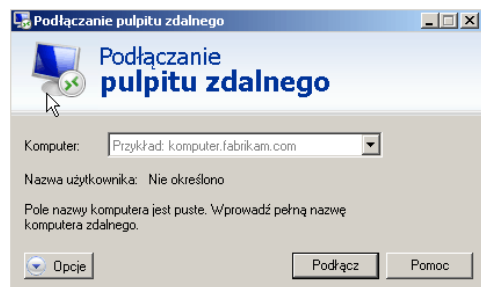
  if(digitalRead(6)==HIGH)
    digitalWrite(11,HIGH);
  else
    digitalWrite(11,LOW);

  if(digitalRead(5)==HIGH)
    digitalWrite(9,HIGH);
  else
    digitalWrite(9,LOW);
}
```





MOŻLIWOŚCI WYKORZYSTANIA STANOWISKA W NAUCZANIU NA ODLEGŁOŚĆ



pulpit zdalny/Team Viewer

Arduino



kamera internetowa



kamera internetowa

Raspbbery Pi



ssh/telnet





1. niskie koszty zakupu Arduino i Raspberry Pi
2. stosunkowo odporna na dewastację konstrukcja platform
3. nie jest wymagany programator
4. darmowe środowiska programistyczne
5. prosta konfiguracja i małe wymagania
6. pozwala skoncentrować uwagę studentów na obsłudze portów wejściowych i wyjściowych oraz urządzeń peryferyjnych
7. bogata oferta urządzeń peryferyjnych
8. możliwość zaimplementowania komunikacji przewodowej (Ethernet lub USB) i bezprzewodowej (WiFi lub Bluetooth).





1. przy zdalnej obsłudze najważniejszym problemem jest brak możliwości modyfikowania konfiguracji stanowiska i układu połączeń
2. wymagane jest stabilne łącze internetowe
3. pracochołonne ustawienie przekierowania portów





111 LAT