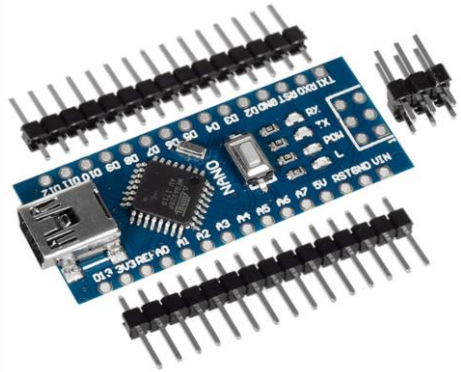
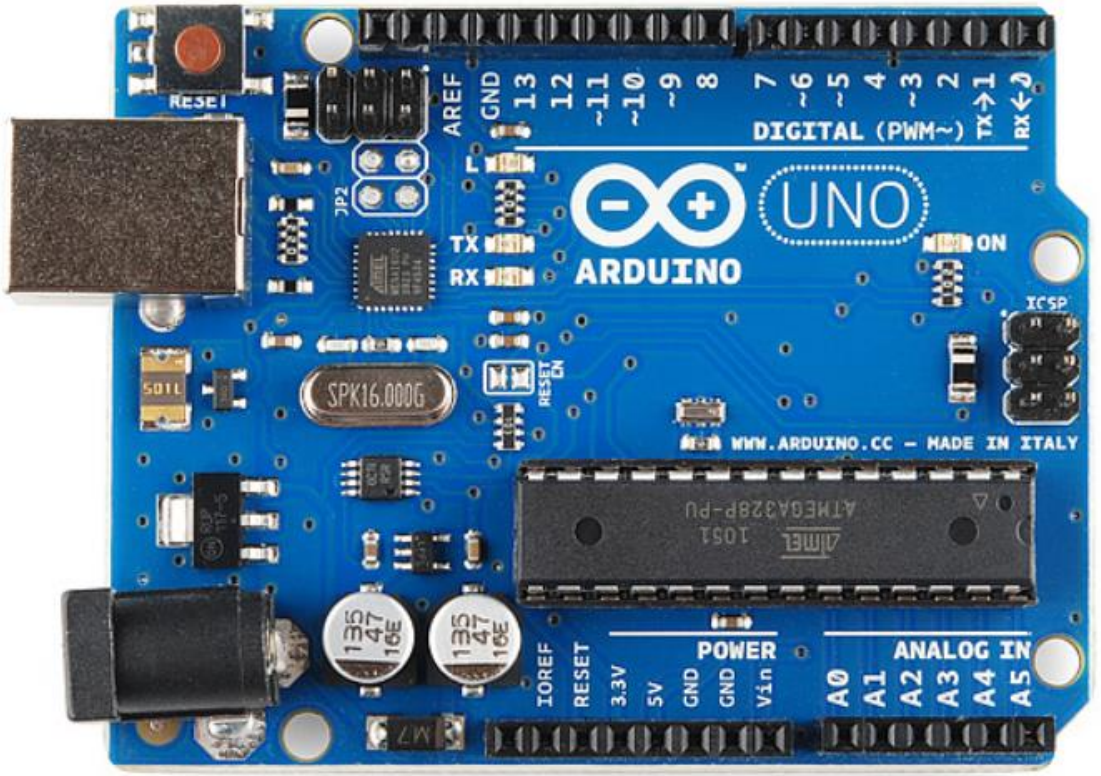
WYŻSZA SZKOŁA INFORMATYKI I ZARZĄDZANIA im. Prof. Tadeusza Kotarbińskiego

SYSTEMY WBUDOWANE

http://ziet.pl/

• mgr inż. Adam Sandomierski – informatyka st. zaoczne 3 rok 2019/2020

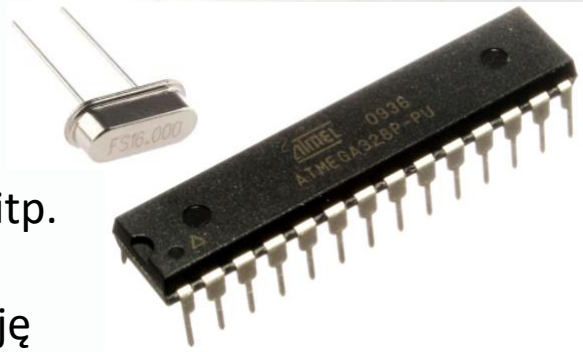


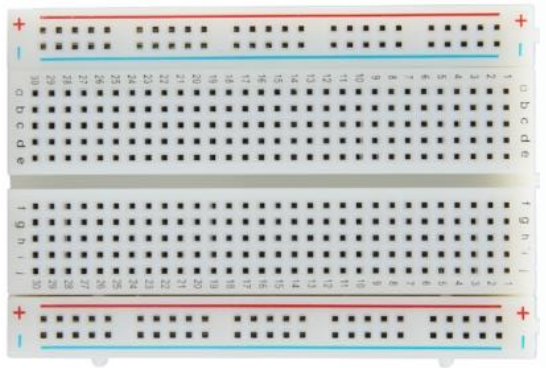


O czym będzie ten przedmiot?
— Programowanie mikrokontrolerów
— Układy Personal Area Network
— Elementy inteligentnego domu
— Połączenie programu ze sprzętem

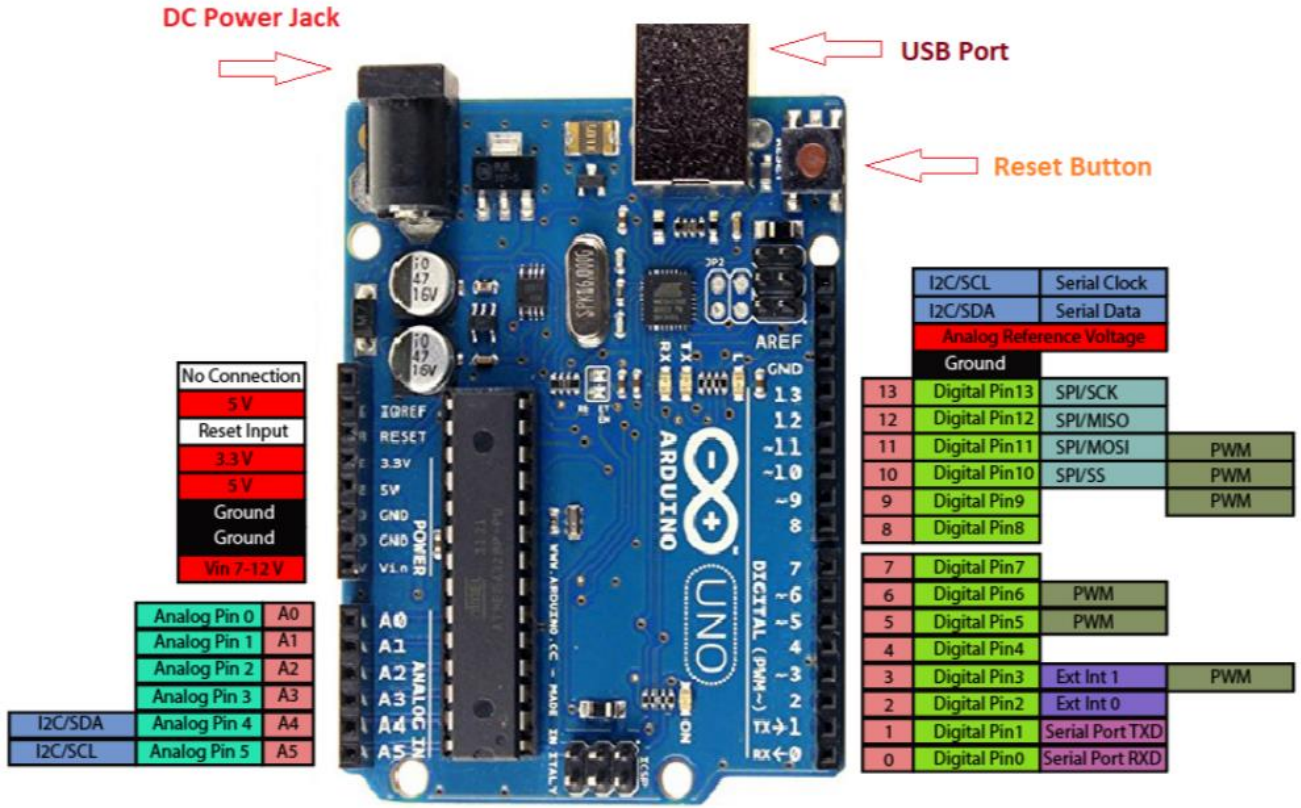
Na czym będziemy bazowali?
— Arduino IDE, Processing, Fritzing
— Arduino/RaspberryPi/…, płytki prototypowe itp. akcesoria
— Elementy elektroniczne, Arduino shields, silniki krokowe itp.

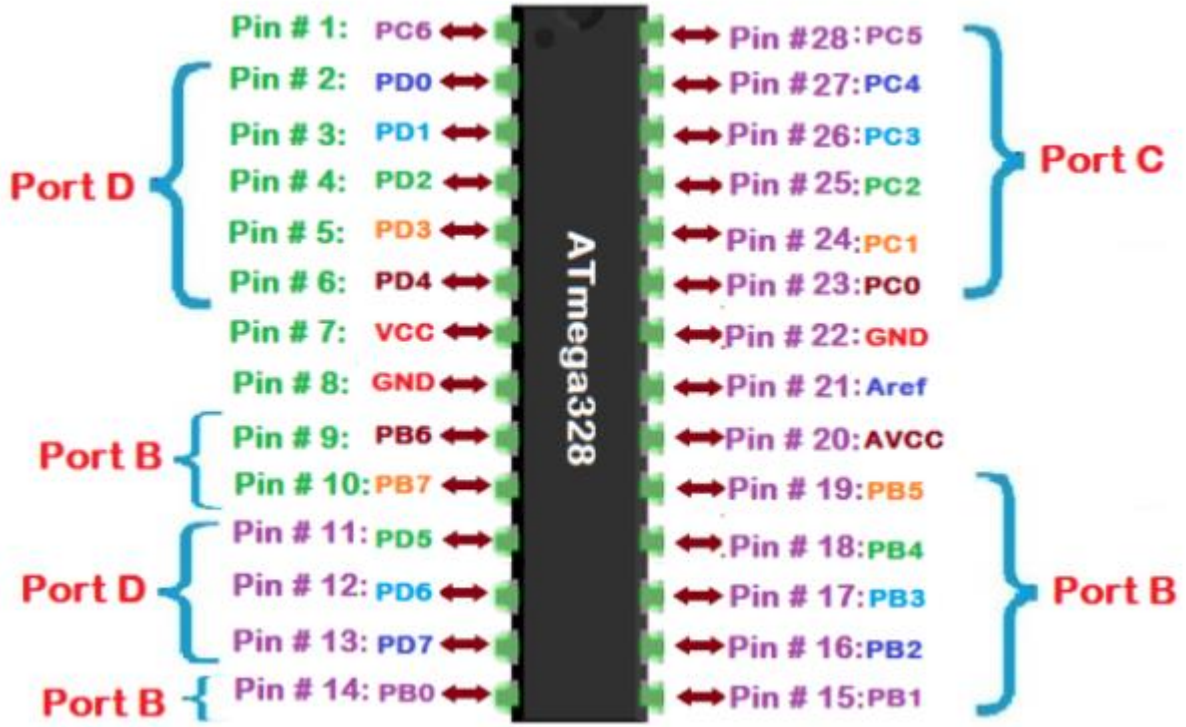
Co chcemy osiągnąć?
— Poznać środowiska Arduino IDE, Scratch oraz Processing
— Opanować podstawy C++/Asm/Java dla Arduino
— Poznać budowę procesora na przykładzie serii Atmega
— Nauczyć się samodzielnej budowy dowolnych układów
… nie bojąc się korzystać z gotowców, transmisji 1wire, I^2C itp.
— Zrealizować indywidualną pracę zaliczeniową, gdzie oprócz pokazania działającego projektu trzeba zrobić dokumentację





Schemat wyprowadzeń (PINOUT) Arduino Uno oraz ATmega328

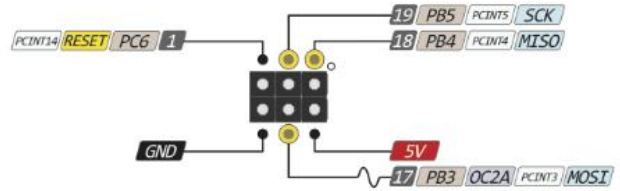


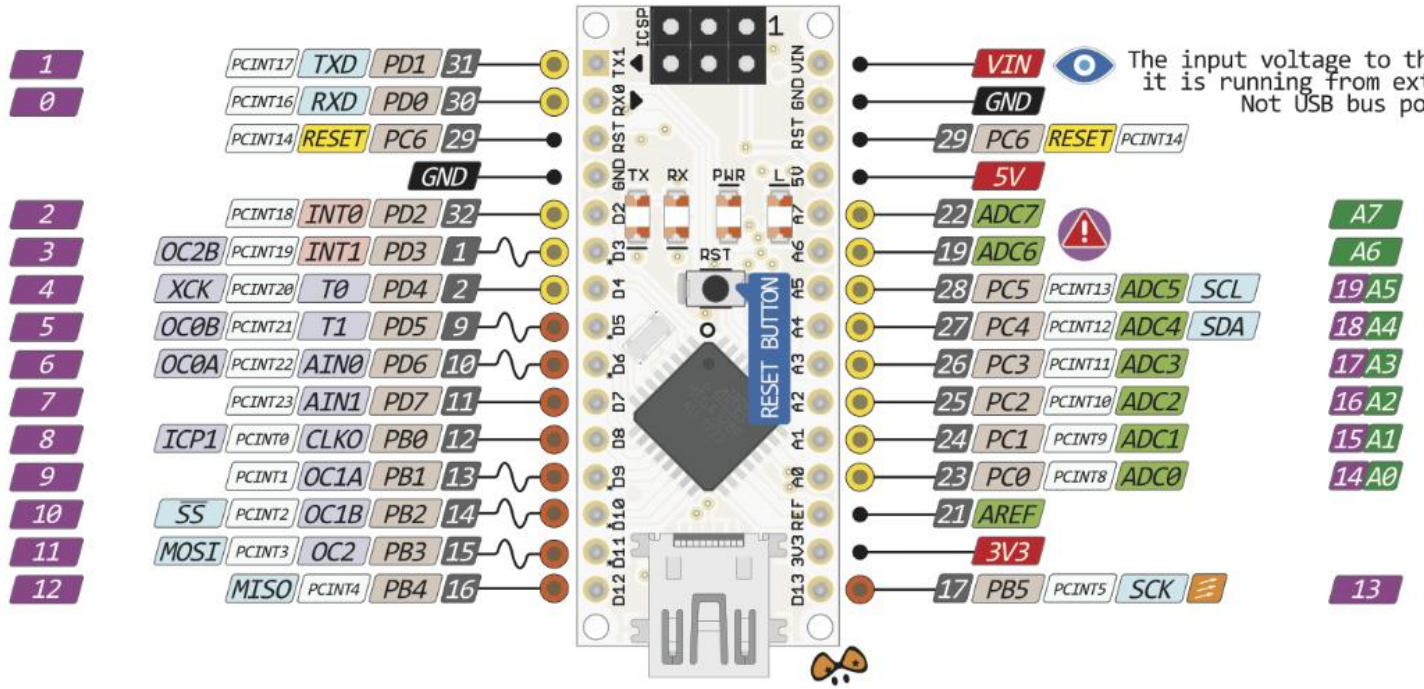


SYSTEMY WBUDOWANE

mgr inż. Adam Sandomierski – informatyka st. zaoczne 3 rok 2019/2020

NANO PINOUT





SYSTEMY WBUDOWANE

mgr inż. Adam Sandomierski – informatyka st. zaoczne 3 rok 2019/2020

Arduino, RaspberryPi, czy inne wynalazki tego typu?

ARDUINO

– działa natychmiast po włączeniu

– zdecydowanie mniejszy pobór energii

– wolniejsza komunikacja (np. Ethernet Shield 10Mbps)

– dużo niższe ceny i możliwość budowy układów CPU+kwarc

– brak GPU i jakichkolwiek złącz do monitorów
brak możliwości obsługi wideo w czasie rzeczywistym

– brak wbudowanego przetwornika D/A (wyjścia analogowego)

RASPBERRY PI

– wymagany system operacyjny, który się uruchamia

– lepsze procesory i możliwość odpalenia Linuksa / Win10

– szybsza komunikacja (wbudowane 1Gbps, WiFi, BLE4.0)

– gotowe rozwiązania All-in-One w kilku wariantach

– wbudowane (µ)HDMI (FHD lub 4k zależnie od wersji)
może zastąpić komputer i robić za SmartTV

– złącza audio (jack stereo) oraz do kamery FHD

– interfejs GPIO (General-Purpose Input/Output) –

– możliwość różnorodnego oprogramowania PINów –

– zasilanie 5V / USB –

Gotowce, czyli jak robić, żeby się nie narobić

– Większość sprzedawców na stronie produktu daje wszystko, co potrzebne, tj.
  – schematy połączeń (które nóżki z czym łączyć)
  – kod przykładowego programu (zwykle trzeba go zrozumieć i nieco przerobić)
  – dokumentację producenta (informacje co więcej dany wynalazek potrafi i jak nim sterować)
  – filmiki do Youtube'a pokazujące jak to może działać w praktyce (najmniej istotne)
– Popularność Arduino jest bardzo duża i wiele osób opisało swoje rozwiązania
– Dużo problemów związanych z elektroniką (np. końcówki mocy na tranzystorach) też jest opisanych
– Jeżeli nie uda się znaleźć czegoś po polsku, to uda się po angielsku ☺

Schemat budowy układu wbudowanego

– Określenie celu: co i kiedy urządzenie ma robić
– Dobranie peryferiów: jaką płytkę, czujniki, shieldy, elementy dyskretne i zasilanie zastosować
– Analiza dostępnych elementów i gotowców
– Zamówienie części, zbieranie bibliotek i projektowanie układu
– Budowa układu na płytce prototypowej, programowanie, testowanie i poprawki
– Projektowanie i wykonanie płytki docelowej
– Udoskonalenie projektu
– Produkcja masowa i sprzedaż ☺

Podstawy programowania dla Arduino (C++)

– Szkic zawiera 2 podstawowe elementy
  void **setup**() { … } – wykonywane 1 raz po włączeniu / resecie
  void **loop**() { … } – wykonywane w nieskończonej pętli

– Dobrym zwyczajem jest nazywanie PINów – dzięki temu późniejsze zmiany można wykonać w 1 miejscu, zamiast w całym kodzie, a dodatkowo wiadomo za co dany PIN odpowiada – służy do tego dyrektywa **#define**

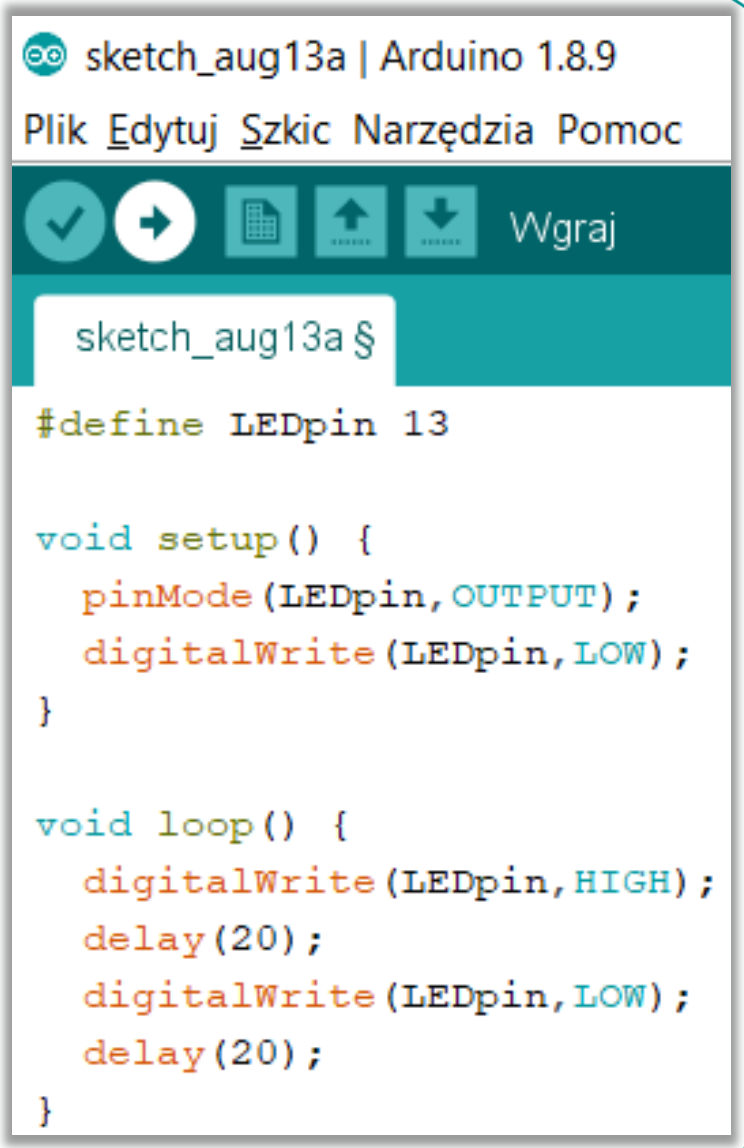
– Dodatkowo na początku programu można definiować zmienne globalne, które będą dostępne we wszystkich definiowanych niżej funkcjach, jak również można definiować procedury (void), funkcje (inne typy), stałe (np. **const int b = 10;**)

– Tryb pracy PINu (od danego momentu w kodzie) określa polecenie
  pinMode(PIN, OUTPUT | INPUT | INPUT_PULLUP)

– Piny oznaczone na schematach tyldą (~) pozwalają na zapis analogowy wartości PWM (Pulse-Width Modulation), można też określić stan cyfrowy PINu
  analogWrite(PIN, 0 … 255) – wartość PWM, 0=0V, 128=50%, 255=5V
  digitalWrite(PIN, 0 | 1 | LOW | HIGH)

– Analogicznie można odczytywać wartości, przy czym PINy analogowe to **A0** … An
  bool cyfrowa = **digitalRead**(PIN) – 1-bit, czyli 0=GND, 1=5V (Vcc)
  int analogowa = **analogRead**(PIN) – 10-bit, czyli 0=GND, 1024=5V

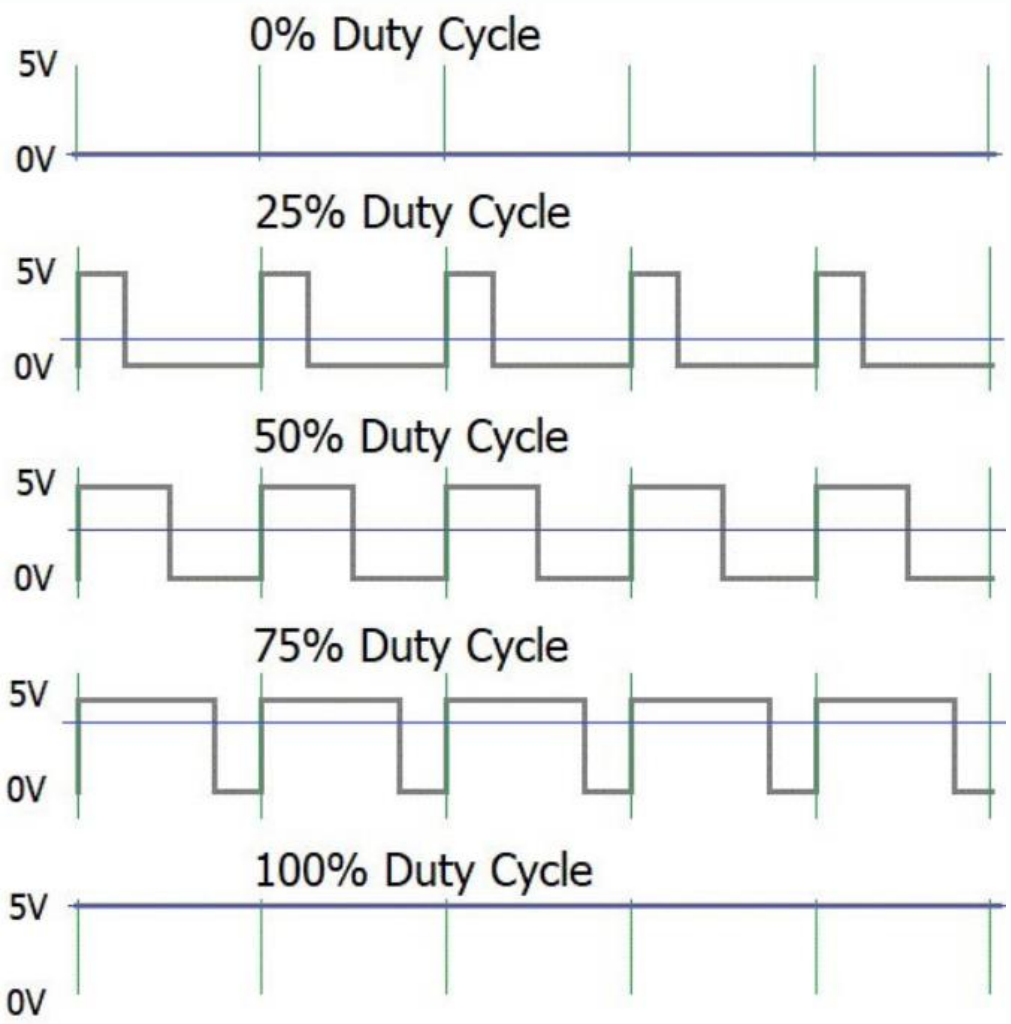
– Opóźnienia można wprowadzać **delay**(ms) lub **delayMicroseconds**(µs)

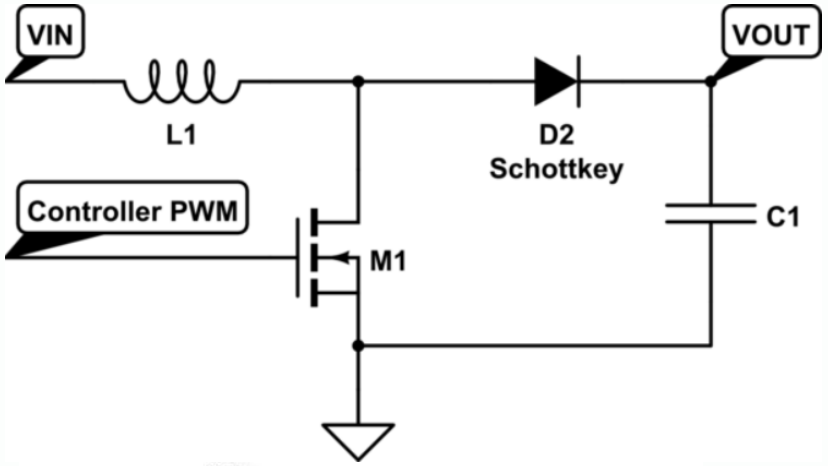


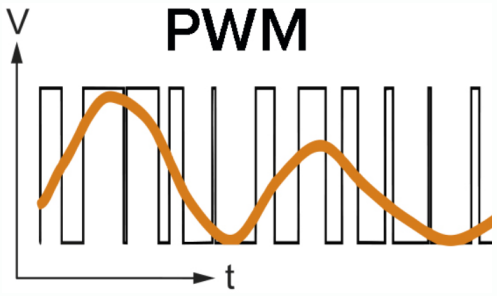
SYSTEMY WBUDOWANE

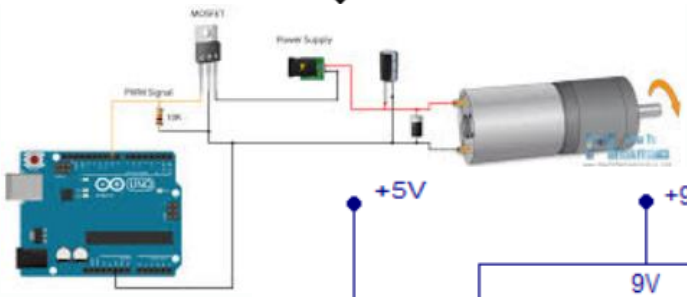
mgr inż. Adam Sandomierski – informatyka st. zaoczne 3 rok 2019/2020

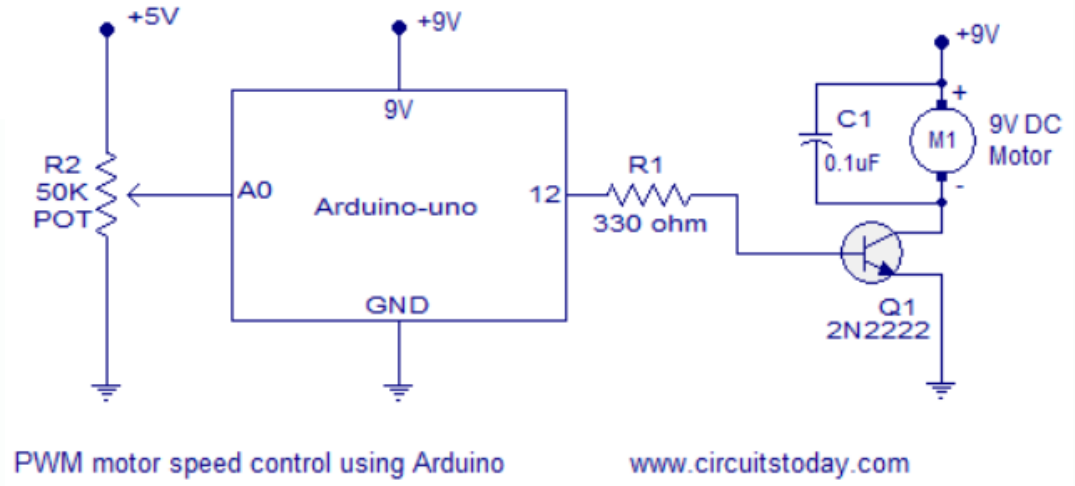
PWM – Pulse-Width Modulation











http://www.energiazero.org/arduino/Arduino%20DC%20Motor%20Control%20Tutorial.pdf

Debugowanie i komunikacja

– Złącze USB może być wykorzystywane jako port szeregowy **Serial**

– Istnieją proste w obsłudze moduły, które pozwalają na komunikację radiową, np.

HC-05 i HC-06	– moduły Bluetooth / UART (jedne z wielu)
nRF24L01	– moduły radiowe 2.4GHz / SPI

– Wśród najpopularniejszych peryferiów można spotkać

DS18B20	– czujnik temperatury 1-wire, zakres -55 do +125 °C
2N2222	– tranzystor bipolarny NPN 40V / 0.8A
LED	– diody elektroluminescencyjne (różne kolory)
220µF / 25V	– kondensator elektrolityczny
NE555	– układ czasowy „timer”
1kΩ, 4k7 (4.7 kΩ)	– rezystory ograniczające prąd wejścia / wyjścia
20kΩ regulowany	– potencjometr o regulowanej rezystancji

WYŻSZA SZKOŁA INFORMATYKI I ZARZĄDZANIA
im. Prof. Tadeusza Kotarbińskiego

Debugowanie i komunikacja

– Ustawienie prędkości
  Serial.begin (kbps)
– Wysłanie znaków
  Serial.print (zmienna)
  Serial.println ("tekst")
– Odczytanie znaków

```
char znak;
while (Serial.available()) {
   znak = (char) Serial.read();
}
```

– Możliwość analogicznej komunikacji z wykorzystaniem Bluetooth, w ten sposób można sterować czymś z telefonu (aplikacją zrobioną np. w środowisku Processing)
– Możliwość komunikacji pomiędzy 2 płytkami Arduino

Monitor portu szeregowego

AnalogInOutSerial

```
  http://www.arduino.cc/en/Tutorial/AnalogInOutSerial
*/

// These constants won't change. They're used to give names to the pins used:
const int analogInPin = A0;  // Analog input pin that the potentiometer is attached to
const int analogOutPin = 9; // Analog output pin that the LED is attached to

int sensorValue = 0;        // value read from the pot
int outputValue = 0;        // value output to the PWM (analog out)

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // map it to the range of the analog out:
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  // change the analog out value:
  analogWrite(analogOutPin, outputValue);

  // print the results to the Serial Monitor:
  Serial.print("sensor = ");
  Serial.print(sensorValue);
  Serial.print("\t output = ");
  Serial.println(outputValue);

  // wait 2 milliseconds before the next loop for the analog-to-digital
  // converter to settle after the last reading:
  delay(2);
}
```