Inteligentne budynki

mgr inż. Jerzy Sobczyk

Arduino

Plan wykładu

- Co to jest Arduino?
- Kontrolery.
- Płytki rozszerzeń.
- Programowanie.
- Przykład.

Własciwości

Arduino składa się z dużego zestawu płytek elektronicznych. Niektóre z nich sa wyposazone w mikrokontrolery, które mogą byc programowane w środowisu Arduino IDE. Użytkownik może łączyć płytki Arduino tworząc nowe urządzenia. Uzytkownik moze dołaczać własne układy elektroniczne zmontowane na własnych płytkach lub płytkach prototypwych Arduino.

- Tanie
- Wieloplatformowe (Windows, Mac OS X, Linux, ChromeOS),
- Przejrzyste iłatwe w obsłudze środowisko programistyczne IDE.
- Rozszerzalne oprogramowanie z otwartym źródłem.
- Rozszerzalny sprzęt z otwartym źródłem.

Arduino Uno

Arduino Uno

ATmega328P microcontroller
14 digital I/O pins
6 PWM digital I/O pins
6 analog input pins
32kB flash
2 kB SRAM
1 kB EEPROM
16 MHz clock

https://www.arduino.cc/en/Main/Products

Arduino Leonardo

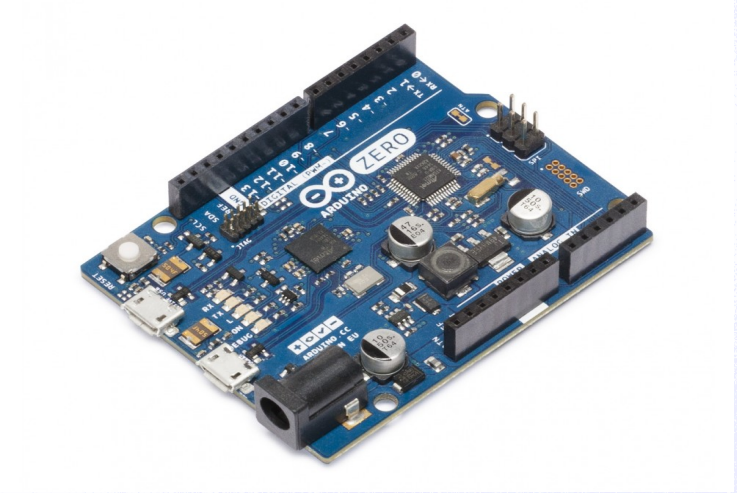
Arduino Leonardo

ATmega32u4 microcontroller
20 digital I/O pins
7 PWM channels
12 analog input channels
32kB flash
2.5 kB SRAM
1 kB EEPROM
16 MHz clock

https://www.arduino.cc/en/Main/Products

Arduino Zero

Arduino Zero



ATSAMD21G18 microcontroller

20 digital I/O pins

10 PWM digital I/O pins

6 analog input pins

1 analog output pin

256kB flash

2 UART

32 kB SRAM

2 channel IC Sound I2S

No EEPROM

Peripheral Touch Controller

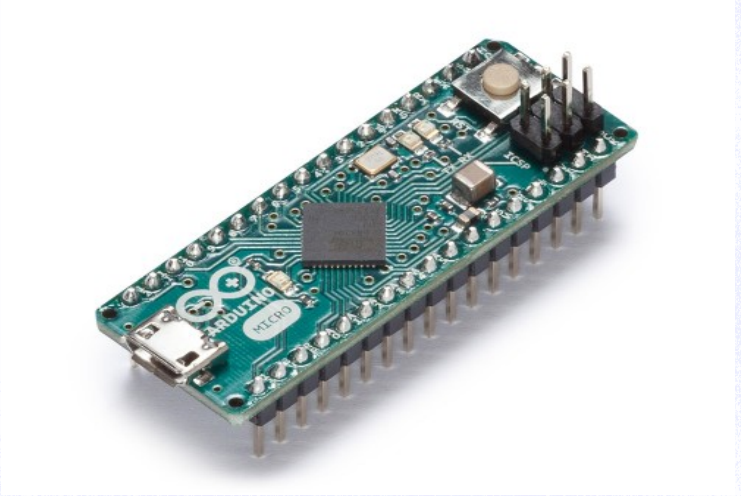
48 MHz clock

32 bit Real Time Counter

https://www.arduino.cc/en/Main/Products

Arduino Micro

Arduino Micro

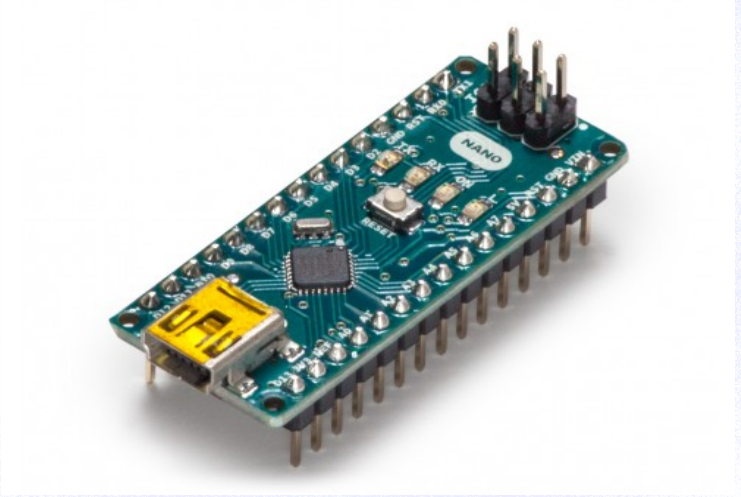


ATmega32U4 microcontroller
20 digital I/O pins
7 PWM channels
12 analog input channels
32kB flash
2.5 kB SRAM
1 kB EEPROM
16 MHz clock

https://www.arduino.cc/en/Main/Products

Arduino Nano

Arduino Nano

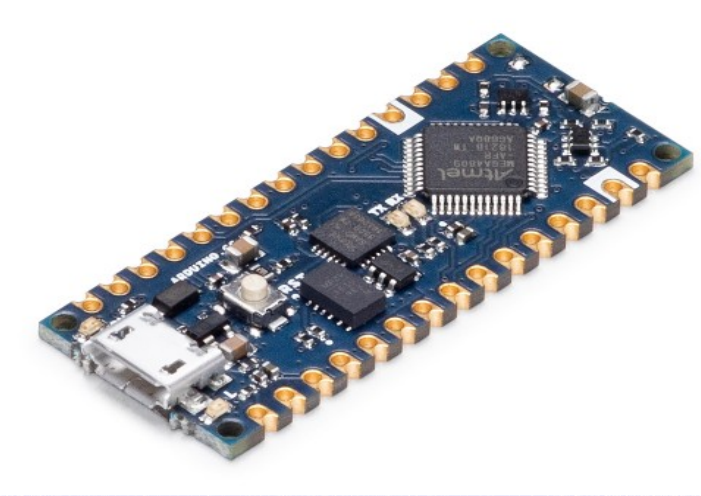


ATmega328 microcontroller
22 digital I/O pins
6 PWM channels
8 analog input channels
32kB flash
2 kB SRAM
1 kB EEPROM
16 MHz clock

https://www.arduino.cc/en/Main/Products

Arduino Nano Every

Arduino Nano Every

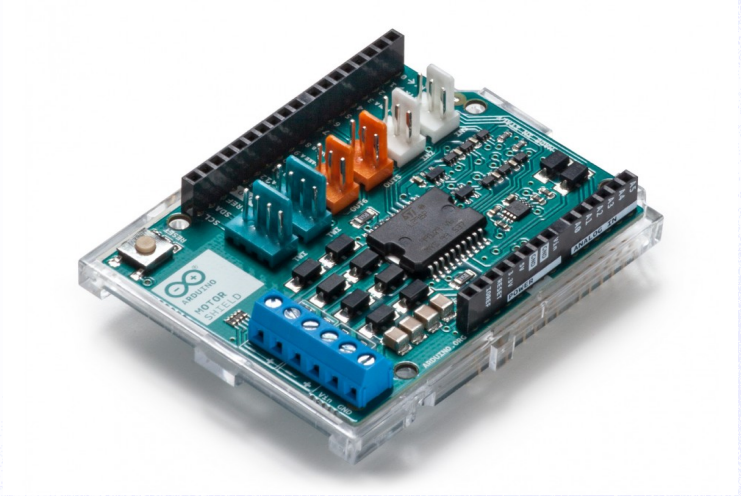


ATmega4809 microcontroller
22 digital I/O pins
5 PWM channels
8 analog input channels
48 kB flash
6 kB SRAM
256 B EEPROM
20 MHz clock

https://www.arduino.cc/en/Main/Products

Arduino Motor

Arduino Motor Shield



two DC motors or one stepper motor

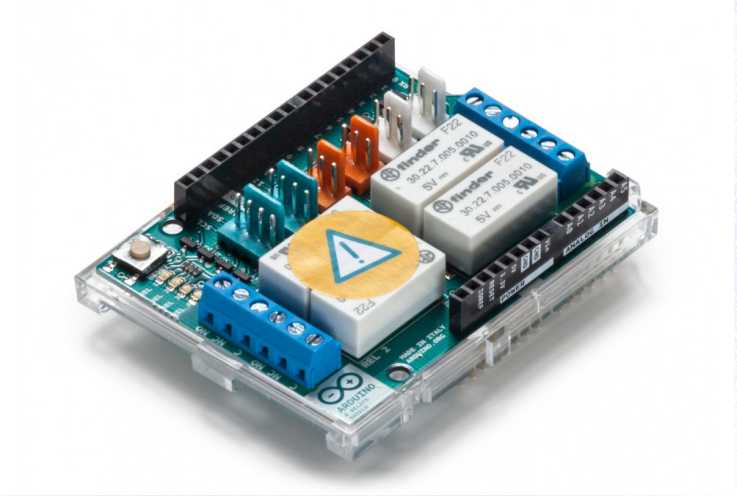
2A max current

L298P motor controller

https://www.arduino.cc/en/Main/Products

Arduino 4 Relays

Arduino 4 Relays Shield



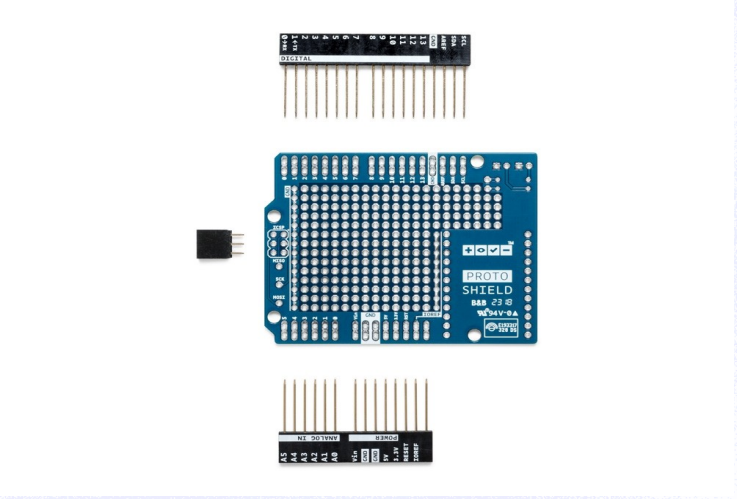
4 relays (60W)

35mA for each realy

https://www.arduino.cc/en/Main/Products

Arduino Prototype

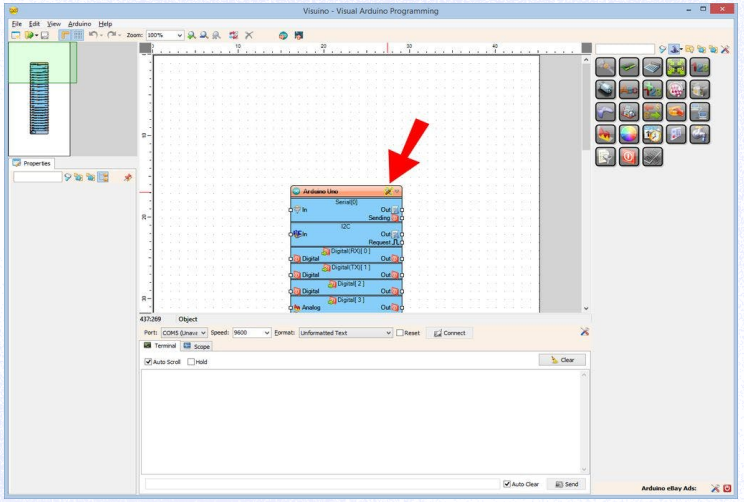
Arduino Proto Shield



https://www.arduino.cc/en/Main/Products

Arduino IDE

Arduino IDE



https://create.arduino.cc/projecthub/mitov/log-gps-information-to-micorsd-card-with-visuino-6b55b6?ref=platform&ref_id=424_trending___&offset=770

Digital and analog I/O

digitalRead(pin) read the value from pin.

digitalWrite(pin,value) write the value to pin.

pinMode(pin,mode) set the pin to INPUT, OUTPU or INPUT_PULLUP mode.

analogRead(pin) read the value from pin.

analogReadResolution(bits) set the resolution returned by analogRead().

analogWriteResolution(bits) set the resolution for analogWrite().

analoglReference(type) set the refernce voltage for analog input.

analoglWrite(pin,value) write the value (PWM wave or DAC) to pin.

Advanced I/O

noTone(pin) stops the generation of square wave started by tone().

tone(pin,frequency,duration) generate square wave of a given frequency and 50% duty cycle.

pulseIn(pin,value,timeout) read the length of a pulse.

pulseInLong(pin,value,timeout) read the length of a pulse. Better for log pulses.

shiftIn(pin,clockPin,bitOrder) read the value from pin and shift it into result.

shiftOut(pin,clockPin,bitOrder,value) shift out one bit .

Time

delay(ms) pauses the program for given amount of miliseconds.

delayMicroseconds(us) pauses the program for given amount of microseconds.

micros() get the number of microseconds sice start of the program. Overflow after approx. 70 minutes.

milis() get the number of miliseconds sice start of the program. Overflow after approx. 50 days.

Keyboard emulation

Keyboard.begin() start emulating the keyboard on the USB port.

Keyboard.end() stop emulating the keyboard on the USB port.

Keyboard.press(key) simulate that key was pressed.

Keyboard.release(key) simulate that key was released.

Keyboard.releaseAll() simulate that all pressed keys were released.

Keyboard.write(char) send one charcter to USB port.

Keyboard.print(char) send one or more charcters to USB port.

Keyboard.print(char) send one or more charcters followed by LF and CR.

Mouse emulation

Mouse.begin() start emulating the mouse on the USB port.

Mouse.end() stop emulating the mouse on the USB port.

Mouse.press(button) simulate that button was pressed.

Mouse.release(button) simulate that button was released.

Mouse.click(button) simulate that button was pressed and released.

Mouse.isPressed(button) check the status of the button.

Mouse.move(x,y,wheel) send changes of the position of the mouse.

Math

abs(val)

constraint(val,min,max)

map(val,fromLow,fromHigh,toLow,toHigh)

max(x,y)

min(x,y)

pow(base,exponent)

sq(val)

sqrt(val)

cos(rad)

sin(rad)

tan(rad)

random(min,max)

randomSeed(seed)

Character

isAlpha(char)

isAlphaNumeric(char)

isAscii(char)

isControl(char)

isDigit(char)

isGraph(char)

isHexadecimalDigit(char)

isLowerCase(char)

isPrintable(char)

isPunct(char)

isSpace(char)

isUpperCase(char)

isWhitespace(char)

Bits and Bytes

bit(n)

bitClear(val,n)

bitSet(val,n)

bitRead(val,n)

highByte(val)

lowByte(val)

Interrupts

attachInterrupt(digitalPinToInterrupt(pin),ISR,mode) attach the Interrupt Service Routine to a pin. Interrupt will be triggered when the pin mode is: LOW, CHANGE, RISING, FALLING, HIGH.

detachInterrupt(digitalPinToInterrupt(pin)) turn of interrupts on a given pin.

interrupts() re-enable interrupts.

noInterrupts() disable interrupts.

Serial communication 1/2

if(Serial) check if Serial port is ready.

Serial.available() get number of bytes received.

Serial.availableForWrite() get number of bytes available for writing.

Serial.begin(speed,config) configure Serial port.

Serial.end(speed,config) disable serial communication.

Serial.find(target,length) read data from the buffer until the target is found.

Serial.findUntil(target,terminator) read data from the buffer until the target ot terminator is found.

Serial.flush() wait for outgoing transmission to complete.

Serial.parseFloat(lookahead,ignore) read floating point number.

Serial.parseInt(lookahead,ignore) read finteger number.

Serial communication 2/2

Serial.peek() read next byte without removing it from the buffer.

Serial.print(val,format) print data to the serial port.

Serial.println(val,format) print data followed by CR and LF to the serial port.

Serial.read() read next byte.

Serial.readBytes(buffer,length) read bytes to the buffer.

Serial.readBytesUntil(char,buffer,length) read bytes until char to the buffer.

Serial.readString() read bytes to the string.

Serial.readString(char) read bytes until char to the string.

Serial.setTimeout(ms) set the timeout to wait for serial data.

Serial.write(buffer,length) writes data from buffer to the serial port.

Serial.serial.Event() function called when data is available.

Stream class 1/2

Stream.available() get number of bytes available in the stream.

Stream.read() read character from the stream.

Stream.flush() clear the buffer after sending.

Stream.find(target,length) read data until the target is found.

Stream.findUntil(target,terminator) read data until the target or terminator is found.

Stream.peek() read a byte wthout removing it.

Stream.readBytes(buffer,length) read characters from the stream.

Stream.readBytesUntil(char,buffer,length) read characters until char from the stream.

Stream class 2/2

Stream.readString() read string from the stream.

Stream.readStringUntil(char) read string until char from the stream.

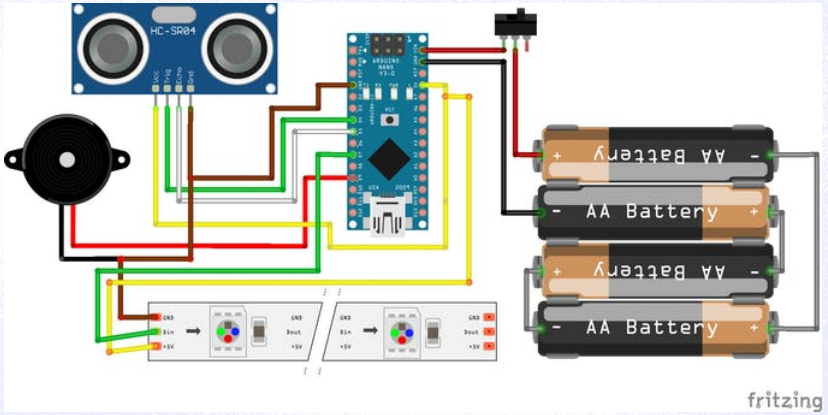
Stream.parseFloat(lookahead,ignore) read floating point number.

Stream.parseInt(lookahead,ignore) read finteger number.

Stream.setTimeout(ms) set the timeout to wait for stream data.

Robot COVID-19 z Arduino Nano

Arduino Nano COVID19 robot



https://create.arduino.cc/projecthub/draakje156/covid-19-simple-friendly-social-distance-robot-watchzi-8f2268?ref=platform&ref_id=424_trending___&offset=6

Program robota COVID-19

```
/*
Friendly Watchzi Social distance robot
Made by Mark Huijbers , Netherlands
Used : Arduino Nano R3
you can change the Neopixel colors ( see color names online)
you can change the tone frequence
*/

#include "FastLED.h"
#define NUM_LEDS 1 // How many leds ,program counting from 0
#define DATA_PIN 7 // Pin7 on NANO (data leds)
CRGB leds[NUM_LEDS];

const int trigPin = 4; //ultransone sensor
const int echoPin = 5; //ultransone sensor
long duration;
int distance;
const int buzzer = 9; // buzzer position

void setup() {

FastLED.addLeds<WS2812B, DATA_PIN, RGB>(leds, NUM_LEDS);
  // type leds Neopixel with controller WS2821B
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(buzzer, OUTPUT);
}

void loop() {


digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance= duration*0.029/2;
  // calculate for using meters , 150 is equal to 1,5 mtr / 6 Feet

if (distance <<= 150){
  leds[0] = CRGB::Red;
  FastLED.show();
  tone(buzzer, 1000); // tone frequence 1 Khz , you can change them
}
else {
   leds[0] = CRGB::Green;
  FastLED.show();
  noTone(buzzer);
}
delay(500); // delay else the neopixel led and measure are to fast
}
```

Powtórzenie

- Co to jest Arduino?
- Dlaczego warto go używać?

Pytania?