Komputerowe wspomaganie eksperymentu
7

Dr Piotr Sitarek

Katedra Fizyki Doświadczalnej, Politechnika Wrocławska

Temat na dziś

Protokół UDP
Arduino

Programowanie w środowisku (cd.)

ni.com

(część materiałów zaczerpnięta ze strony producenta)

Protokół UDP

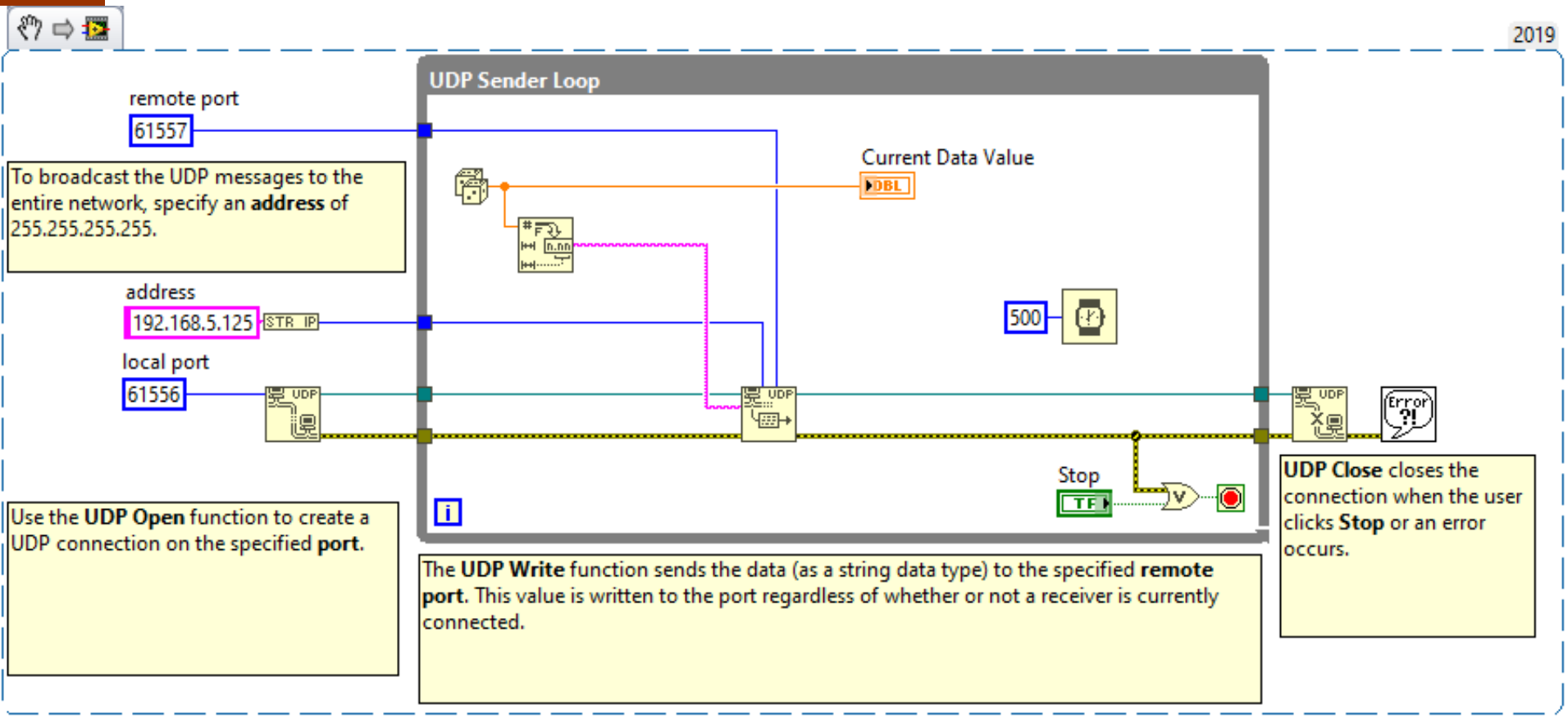
- **UDP** (ang. *User Datagram Protocol* – protokół pakietów użytkownika) – jeden z protokołów internetowych. Nie gwarantuje dostarczenia datagramu.
- Jest to protokół bezpołączeniowy, więc nie ma narzutu na nawiązywanie połączenia i śledzenie sesji (w przeciwieństwie do TCP).
- Nie ma też mechanizmów kontroli przepływu i retransmisji.
- Korzyścią płynącą z takiego uproszczenia budowy jest szybsza transmisja danych i brak dodatkowych zadań, którymi musi zajmować się host posługujący się tym protokołem.
- Z tych względów UDP jest często używany w takich zastosowaniach jak wideokonferencje, strumienie dźwięku w Internecie i gry sieciowe, gdzie dane muszą być przesyłane możliwie szybko. Przykładem może być VoIP lub protokół DNS.

Protokół UDP

+	Bity 0 – 7	8 – 15	16 – 23	24 – 31
0	Adres źródłowy			
32	Adres docelowy			
64	Zera	Protokół	Długość UDP	
96	Port źródłowy		Port docelowy	
128	Długość		Suma kontrolna	
160	Dane			

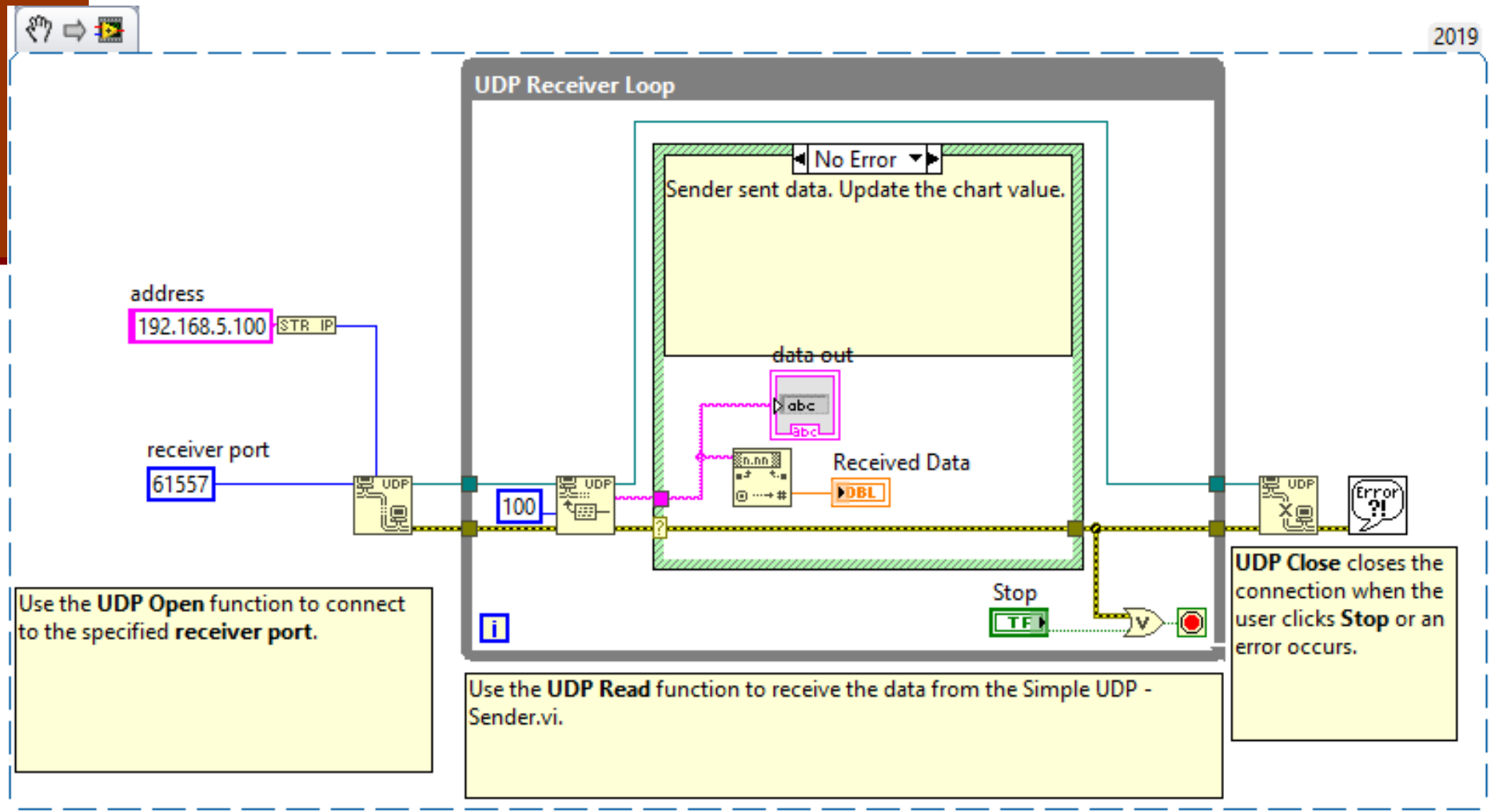
Protokół UDP

Pakiet „Simple UDP.lvproj”



Protokół UDP

Pakiet „Simple UDP.lvproj”



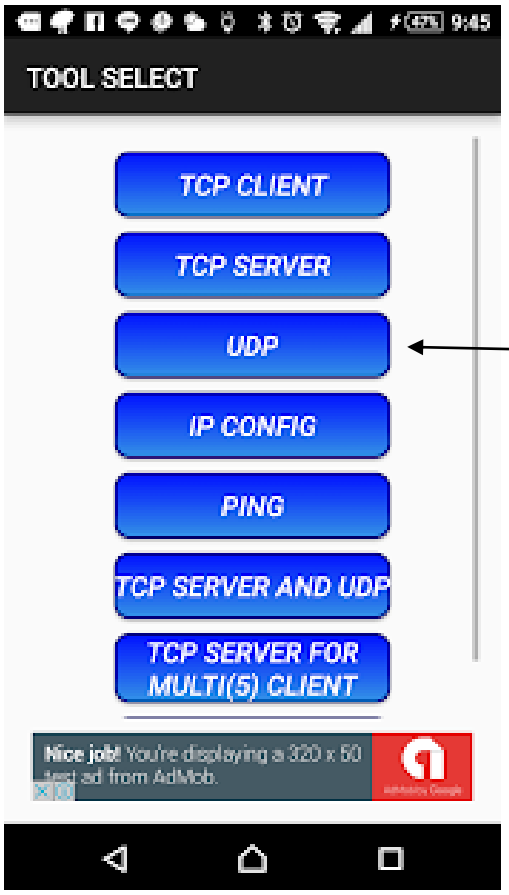
Protokół UDP

Przykładowa aplikacja na smartfon:

TCP/UDP TEST TOOL

Protokół UDP

Przykładowa aplikacja na smartfon:



wybieramy UDP

Protokół UDP

Przykładowa aplikacja na smartfon:

takie samo IP i nr portu powinny być w programie odbierającym dane z telefonu

adres IP komputera

port komputera

ustanawiamy połączenie

wpisujemy dane

wysyłamy dane, przy odbieraniu danych powinno być wyłączone

wysyłanie ciągłe

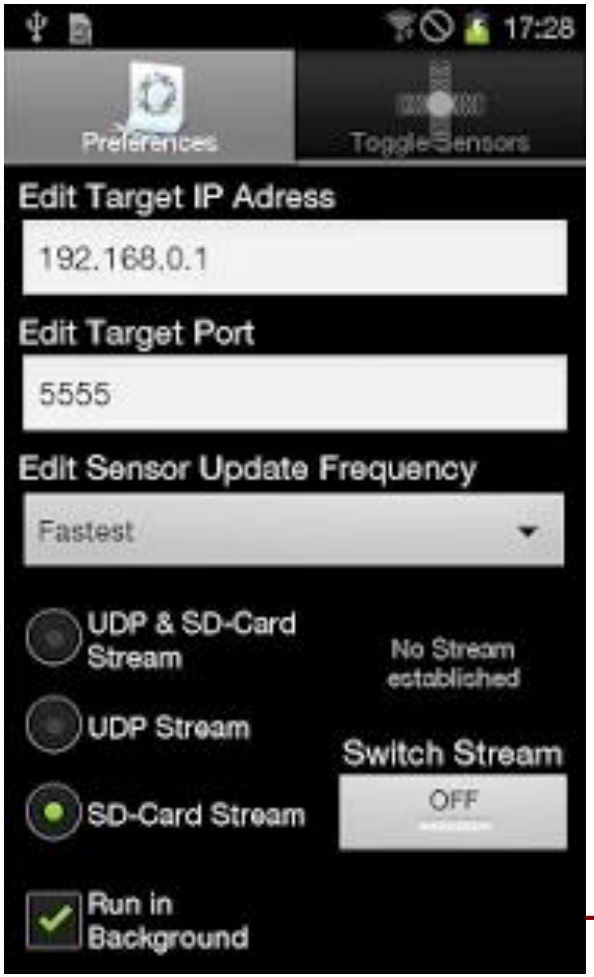
Protokół UDP

Przykładowa aplikacja na smartfon:

Sensorstream IMU+GPS

Protokół UDP

Przykładowa aplikacja na smartfon:



adres IP komputera

port

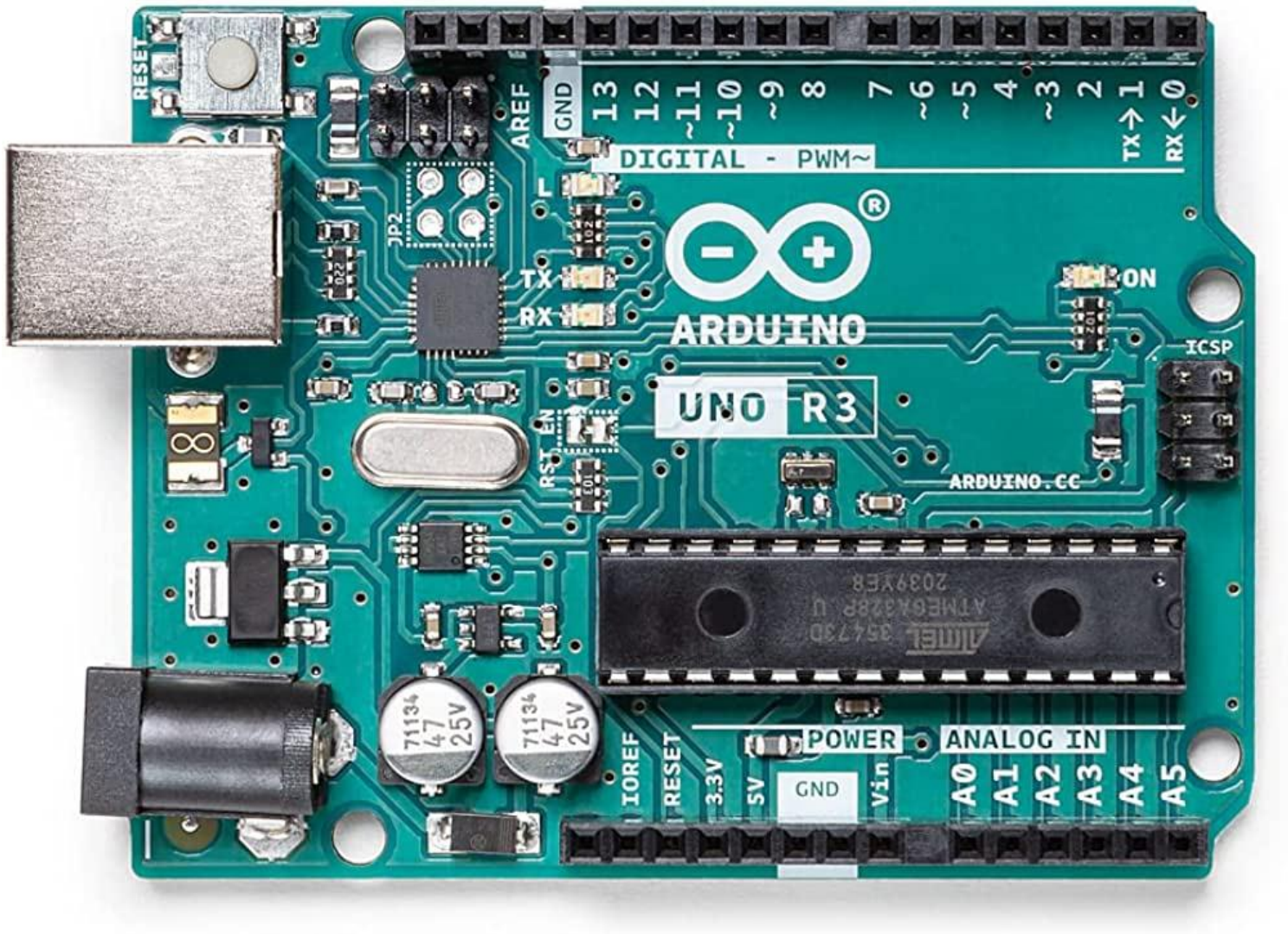
Protokół UDP

Przykładowa aplikacja na smartfon:

Zadania

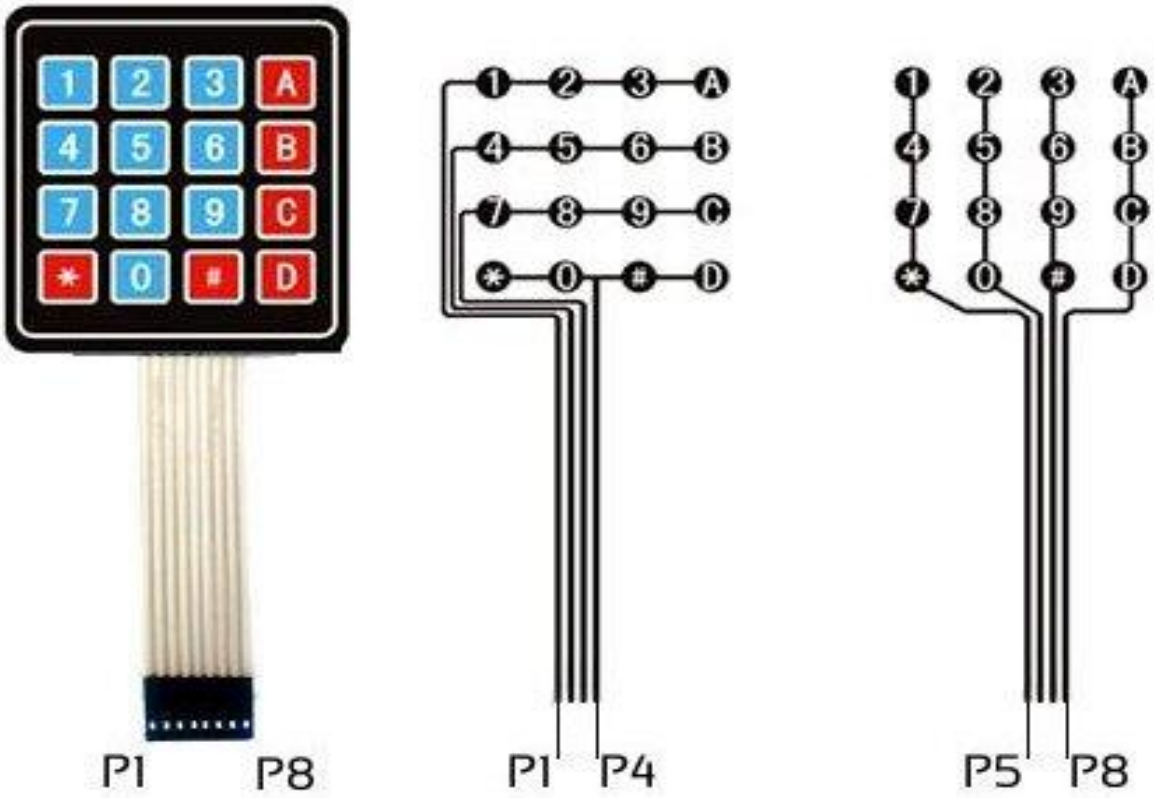
- znaleźć i uruchomić programy (Pakiet „Simple UDP.lvproj”):
  Simple UDP - Sender.vi
  Simple UDP - Receiver.vi
- uruchomić „Sender” oraz „Receiver”
- zainstalować na telefonie aplikację „TCP/UDP TEST TOOL” skonfigurować program – w zakładce UDP wpisać IP komputera i nr portu – uruchomić „Connect”
- w programie „Sender” wpisać IP i port telefonu – uruchomić (dane powinny być przesyłane z komputera na telefon)
- uruchomić i skonfigurować program „Receiver”
- przesłać dane z telefonu na komputer
- na smartfonie zainstalować aplikację (np.) „Sensorstream IMU+GPS”
- skonfigurować program „Receiver” i aplikację
- przesłać dane ze smartfonu na komputer
- uzupełnić program – odczytać i wykorzystać/zaprezentować przesłane dane

Arduino


RESET
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL - PWM~
L
TX
RX
ON
ARDUINO
UNO R3
ICSP
RST EN
ARDUINO.CC
ATMEGA328P
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
Vin
A0
A1
A2
A3
A4
A5

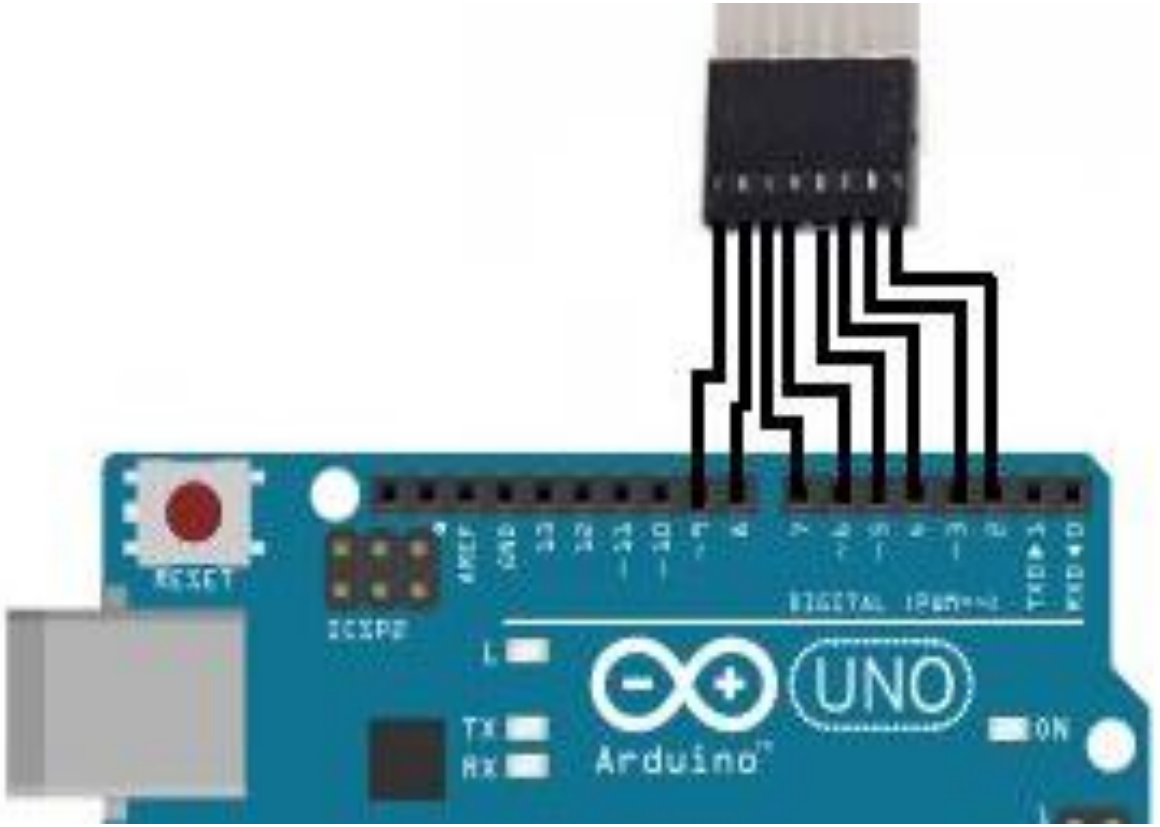
Arduino - keypad

- **Klawiatura membranowa 16 klawiszy (4x4)**



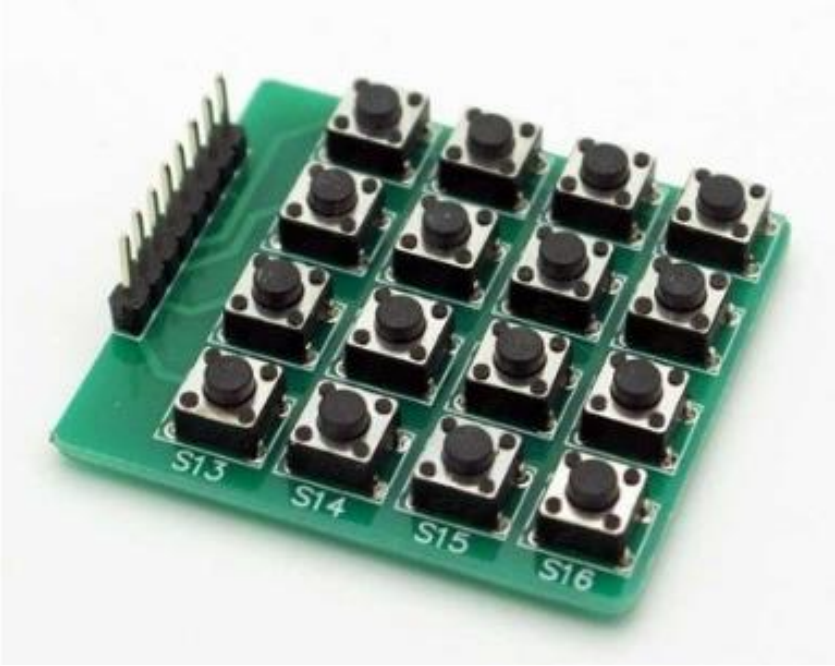
Arduino - keypad

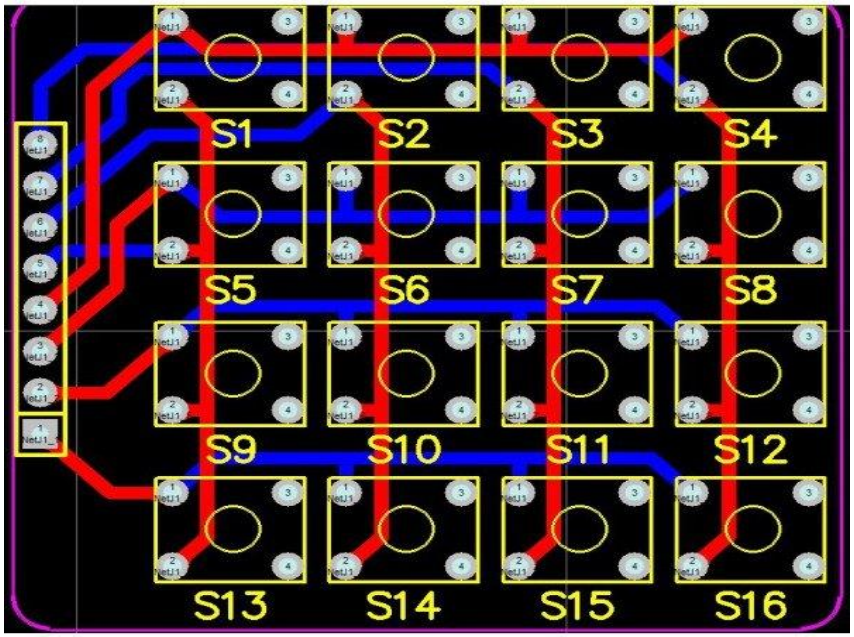
- **Klawiatura membranowa 16 klawiszy (4x4)**



Arduino - keypad

- **Klawiatura TACT 4x4 switch**





keypad1.ino

```
/*4x4 Matrix Keypad connected to Arduino
This code prints the key pressed on the keypad to the serial port*/

#include <Keypad.h>
const byte numRows= 4; //number of rows on the keypad
const byte numCols= 4; //number of columns on the keypad
//keymap defines the key pressed according to the row and columns just as appears on the keypad
char keymap[numRows][numCols]=
{
{'1', '2', '3', 'A'},
{'4', '5', '6', 'B'},
{'7', '8', '9', 'C'},
{'*', '0', '#', 'D'}
};
//Code that shows the the keypad connections to the arduino terminals
byte rowPins[numRows] = {9,8,7,6}; //Rows 0 to 3
byte colPins[numCols]= {5,4,3,2}; //Columns 0 to 3

//initializes an instance of the Keypad class
Keypad myKeypad= Keypad(makeKeymap(keymap), rowPins, colPins, numRows, numCols);
```

keypad1.ino

```
/*4x4 Matrix Keypad connected to Arduino
This code prints the key pressed on the keypad to the serial port*/


void setup()
{
Serial.begin(9600);
}


//If key is pressed, this key is stored in 'keypressed' variable
//If key is not equal to 'NO_KEY', then this key is printed out
//if count=17, then count is reset back to 0 (this means no key is pressed during the whole keypad scan process
void loop()
{
char keypressed = myKeypad.getKey();
if (keypressed != NO_KEY)
{
Serial.print(keypressed);
}
}
```

Keypad by Mark Stanley, Alexander Brevig Wersja 3.1.1 **INSTALLED**

Keypad is a library for using matrix style keypads with the Arduino. As of version 3.0 it now supports mulitple keypresses. This library is based upon the Keypad Tutorial. It was created to promote Hardware Abstraction. It improves readability of the code by hiding the pinMode and digitalRead calls for the user.

More info

Arduino - joystick

- **Joystick analogowy X/Y z przyciskiem - 5V do Arduino**

Arduino - joystick

- **Joystick analogowy X/Y z przyciskiem - 5V do Arduino**

GND
+5V
VRX A0
VRY A1
SW A2

Arduino - joystick

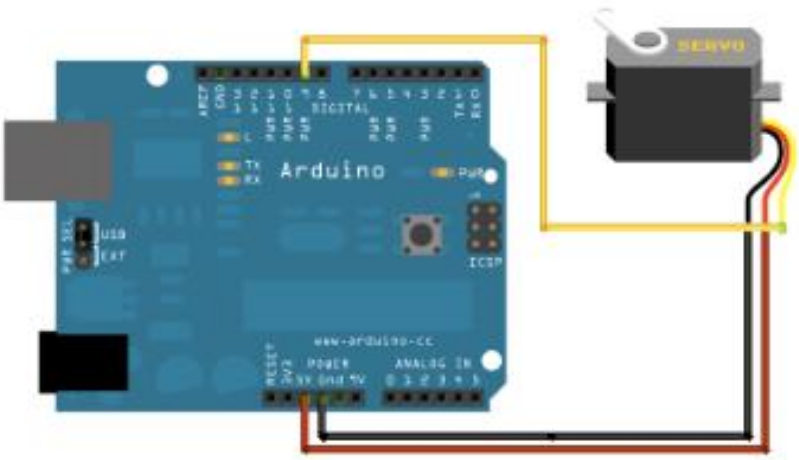
- **Joystick analogowy X/Y z przyciskiem - 5V do Arduino**

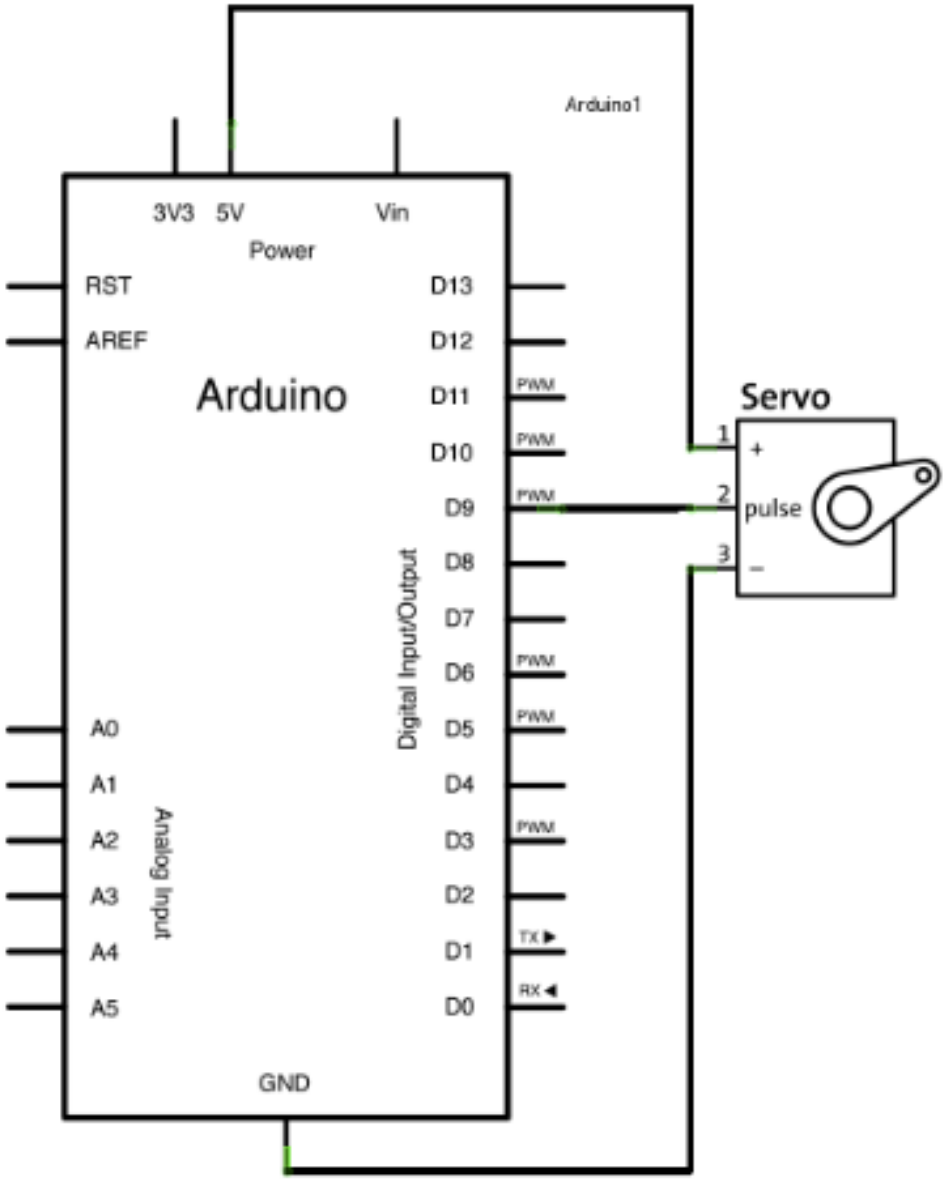
joystick.ino

Podgląd – kreślarka pod A. IDE

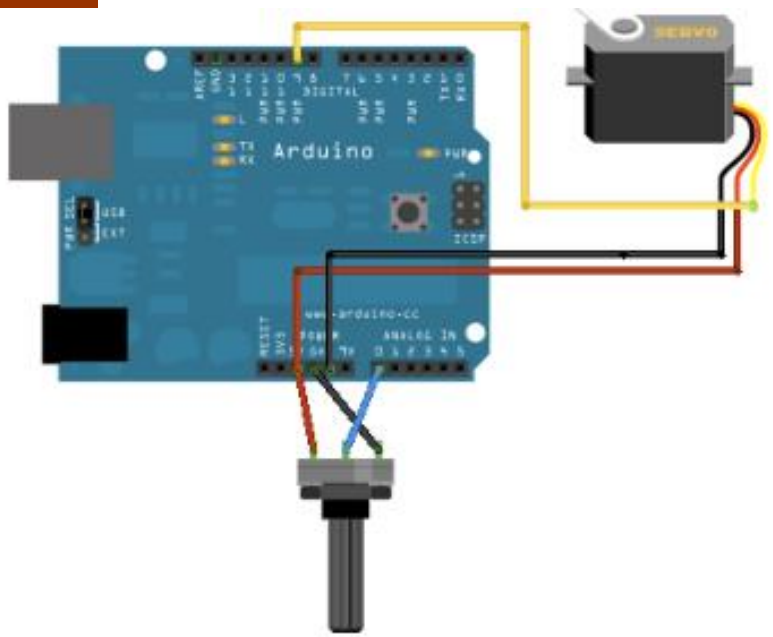
Arduino - servo

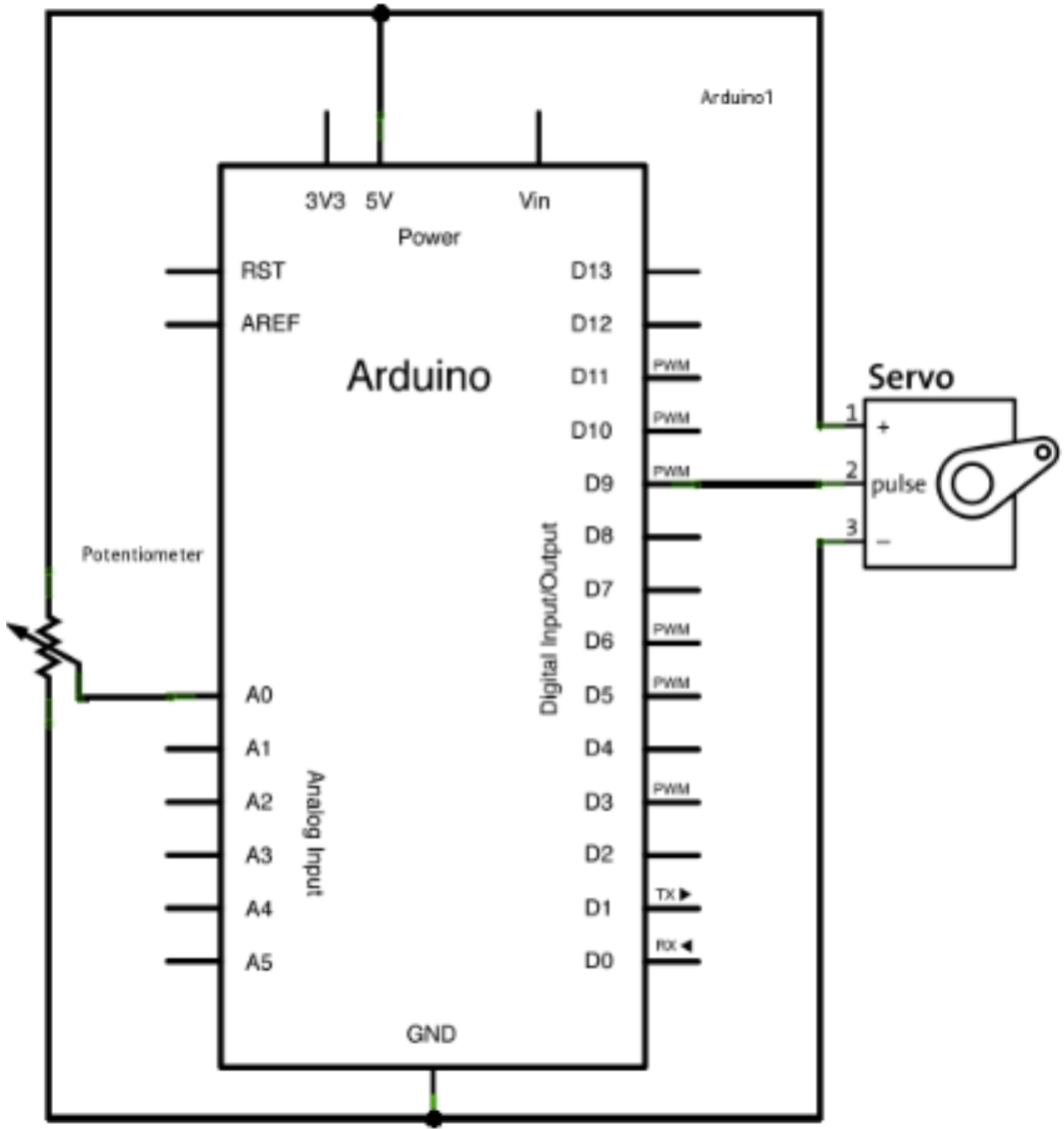
Arduino - servo





Arduino - servo



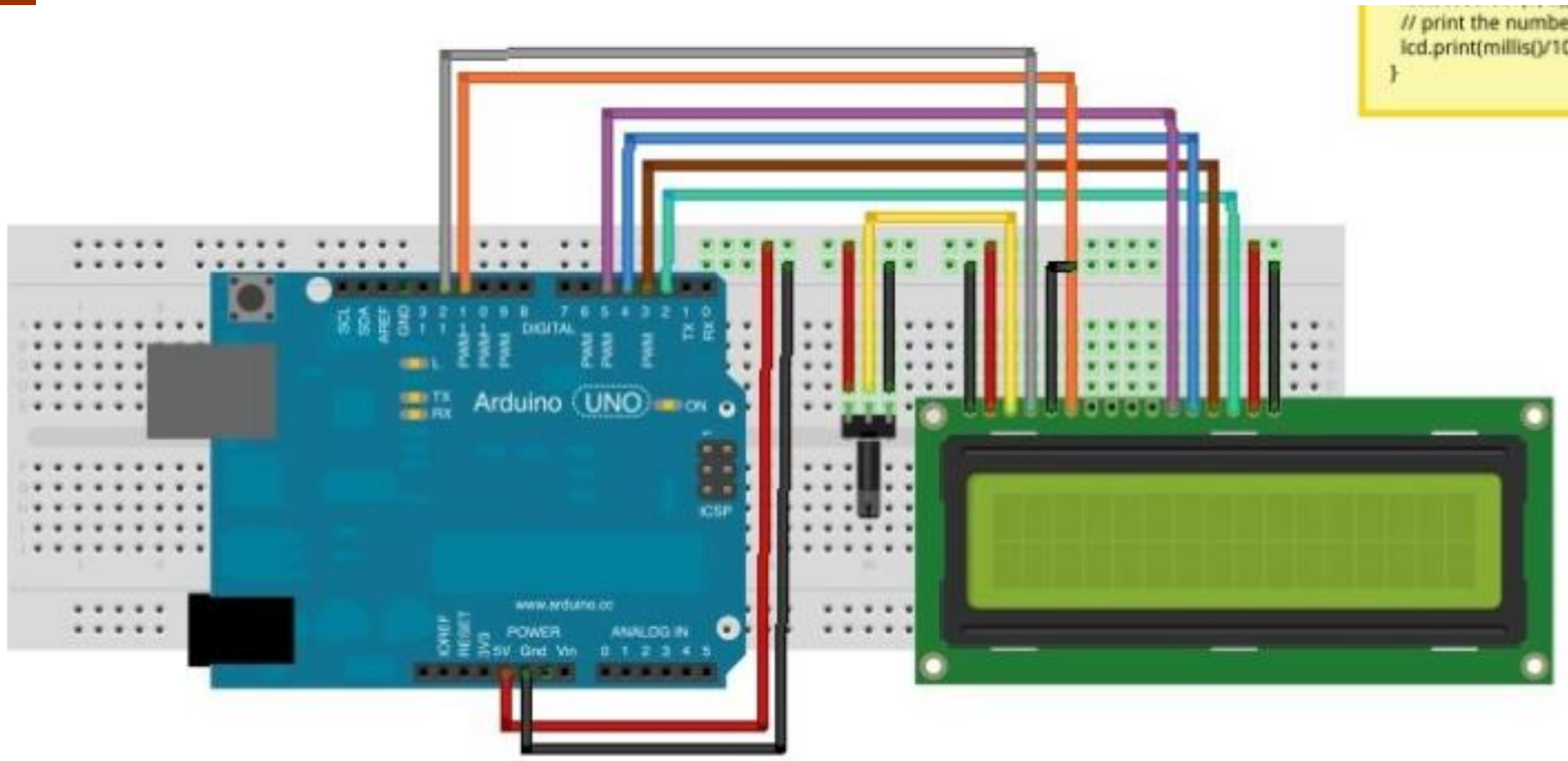


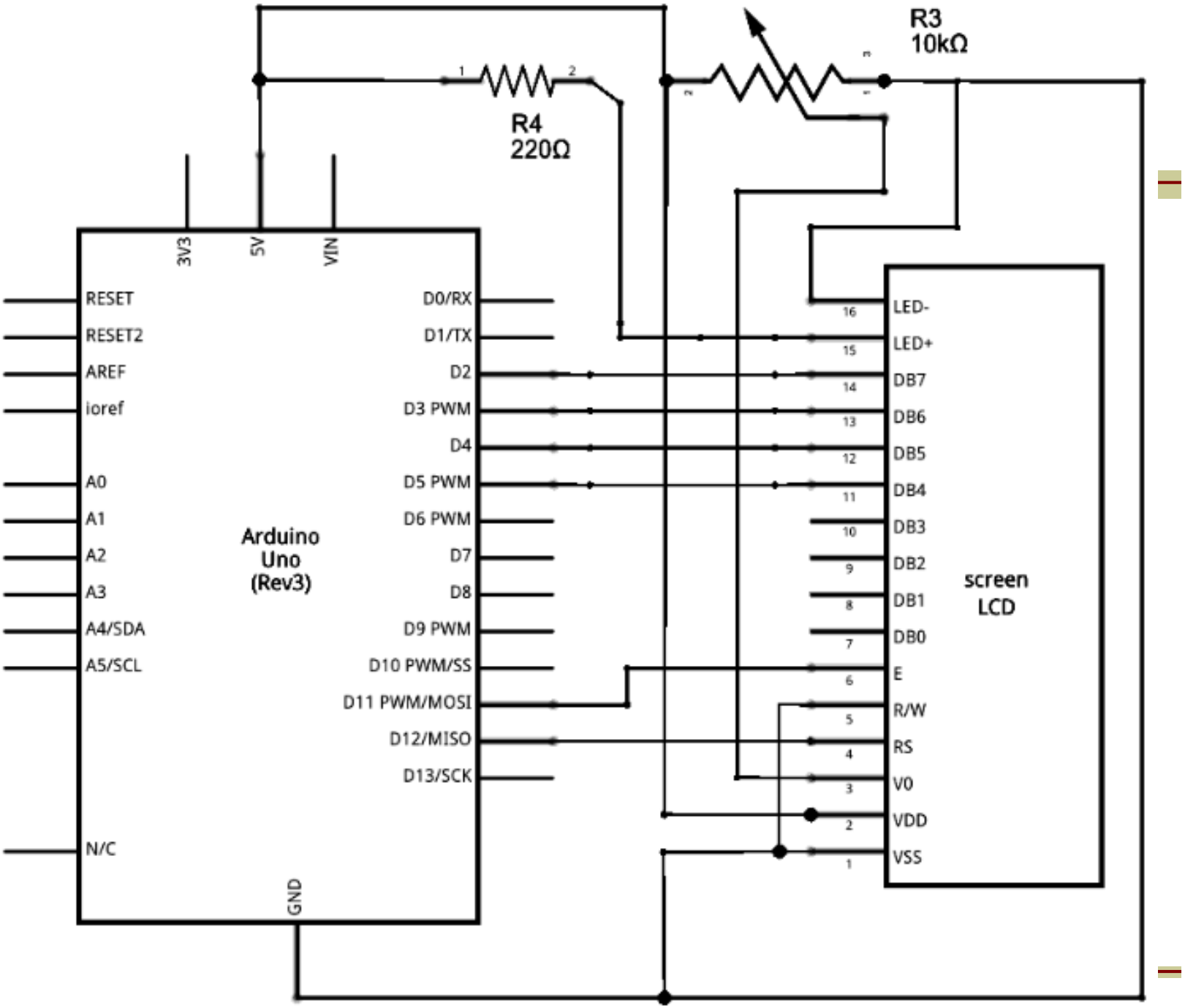
Arduino - wyświetlacz

- **Wyświetlacz LCD 2x16 niebieski ze sterownikiem HD44780 - QC1602A**

Arduino - wyświetlacz

- **Wyświetlacz LCD 2x16 niebieski ze sterownikiem HD44780 - QC1602A**




R3
10kΩ
R4
220Ω
3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
LED-
LED+
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
E
R/W
RS
V0
VDD
VSS
screen
LCD

Arduino - wyświetlacz

- **Wyświetlacz LCD 2x16 niebieski ze sterownikiem HD44780 - QC1602A**

1. VSS Masa
2. VDD Zasilanie +5V
3. V0 Kontrast
4. RS Wybór rejestru instrukcji wyświetlacza (stan niski) albo rejestru danych(wysoki)
5. R/W Odczyt (stan niski) / Zapis (stan wysoki)
6. E Odblokowanie wyświetlacza
7. DB0
8. DB1
9. DB2
10. DB3
11. DB4
12. DB5
13. DB6
14. DB7
15. LED A Zasilanie podświetlania +5V
16. LED K Masa podświetlania