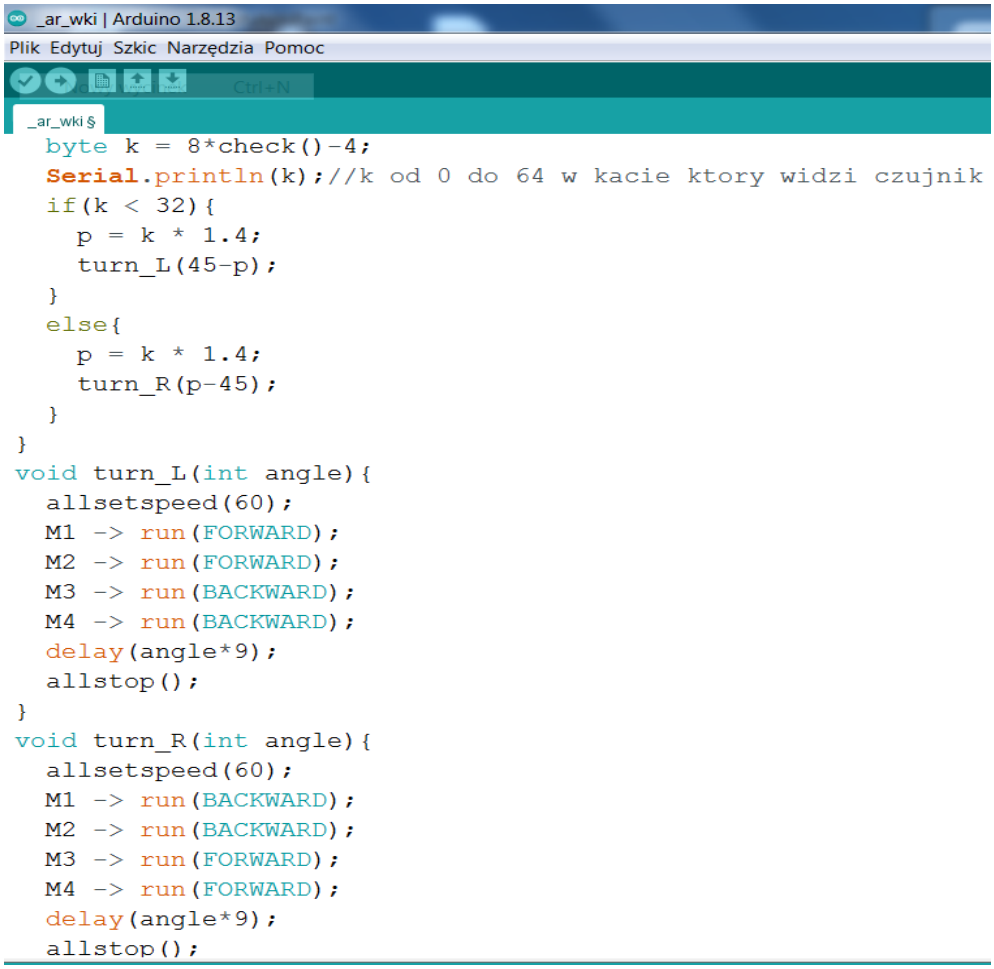
_ar_wki §

```
#include <Wire.h>
#include <Adafruit_MotorShield.h>
#include "utility/Adafruit_MS_PWMServoDriver.h"
#include <Stepper.h>
#define button 0
#define trigPin A1
#define echoPin A0
Adafruit_MotorShield AFMS = Adafruit_MotorShield();
Adafruit_DCMotor *M1 = AFMS.getMotor(1);
Adafruit_DCMotor *M2 = AFMS.getMotor(2);
Adafruit_DCMotor *M3 = AFMS.getMotor(3);
Adafruit_DCMotor *M4 = AFMS.getMotor(4);
Stepper stepper (32, 8, 10, 9, 11);
bool stan = false;
int bt = 1;
void setup() {
  AFMS.begin();
  Serial.begin(9600);
  randomSeed(analogRead(A5));
  pinMode(button, INPUT_PULLUP);
}
void loop() {
  if(Serial.available() > 0){
      bt = Serial.read() - 48;
  }
    switch (bt){
      case 0:
```

Otwórz

_ar_wki §

```
      case 0:
        if(avg_distance() < 25) //srednia z ostatnich 10 pomiarów mniejsza od 15cm to skręć
          newposition();
        fw(150);
        break;
      case 2:     //klawiatura 2,4,5,6,8 to strzałki
        fw(150);
        break;
      case 8:
        bw(150);
        break;
      case 4:
        turn_L(45);
        break;
      case 6:
        turn_R(45);
        break;
      case 5:
        allstop();
        break;
    }
}
void newposition(){
  allstop();
  delay(500);
  turn_L(random(90, 270));
  float p;              // zmienna pomocnicza, żebyb zamiienic niecałkowity kat na calkowity
  byte k = 8*check()-4;
```


_ar_wki | Arduino 1.8.13
Plik Edytuj Szkic Narzędzia Pomoc
Ctrl+N
_ar_wki §
byte k = 8*check()-4;
Serial.println(k);//k od 0 do 64 w kacie ktory widzi czujnik
if(k < 32){
p = k * 1.4;
turn_L(45-p);
}
else{
p = k * 1.4;
turn_R(p-45);
}
}
void turn_L(int angle){
allsetspeed(60);
M1 -> run(FORWARD);
M2 -> run(FORWARD);
M3 -> run(BACKWARD);
M4 -> run(BACKWARD);
delay(angle*9);
allstop();
}
void turn_R(int angle){
allsetspeed(60);
M1 -> run(BACKWARD);
M2 -> run(BACKWARD);
M3 -> run(FORWARD);
M4 -> run(FORWARD);
delay(angle*9);
allstop();

_ar_wki §

```
  allstop();
}
void allsetspeed(byte v){
  M1 -> setSpeed(v);
  M2 -> setSpeed(v);
  M3 -> setSpeed(v);
  M4 -> setSpeed(v);
}
void allstop(){
  M1 -> run(RELEASE);
  M2 -> run(RELEASE);
  M3 -> run(RELEASE);
  M4 -> run(RELEASE);
}
void fw (byte v){
  allsetspeed(v);
  M1 -> run(FORWARD);
  M2 -> run(FORWARD);
  M3 -> run(FORWARD);
  M4 -> run(FORWARD);
}
void bw (byte v){
  allsetspeed(v);
  M1 -> run(BACKWARD);
  M2 -> run(BACKWARD);
  M3 -> run(BACKWARD);
  M4 -> run(BACKWARD);
}
```

_ar_wki §

```
int distance (){
  long t = 0, s = 0;
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  t = pulseIn(echoPin, HIGH);
  s = t/58;
  return s;
}
int avg_distance(){
  int d[10] = {0}, suma = 0;
  for(int i = 0; i < 10; i++){
    d[i] = distance();
    suma = suma + d[i];
    delay(10);
  }
  return suma / 10;
}
byte check(){     //stworz 64 elementowa tablice z odleglosciami od przeszkod, zwroc indeks tablicy gdzie odleglosc jest najwieksza
  byte i = 0;
  int dist[64] = {0};
  stepper.setSpeed(500);
  stepper.step(-256);
  for(i = 0; i < 64; i++){
    dist[i] = distance();
    stepper.step(8);
```

```
    stepper.step(8);
  }
  stepper.step(-256);
  return findmax_id(dist);
}
byte findmax_id(int dist[]){        //znajdz indeks w ktorym odleglosc od przeszkody jest najwieksza
  byte max_id = 0;
  long sum[8] = {0};
  for( byte i = 0; i < 8; i++){
    for( byte j = 0; j < 8; j++){
      sum[i] = sum[i] + dist[i+(i-1)*j];
    }
    sum[i] = sum[i] / 8;
    if( sum[i] > sum[max_id] )
      max_id = i;
  }
  return max_id;
}
```