

Załącznik 1

//////////////////////////////////ZMIENNE//////////////////////////////////

#include <FlexiTimer2.h>

#include <Servo.h>

#include <Wire.h>

#define resistor 1000

#define ledpin A3

#define seriesresistor 10000

#define SLAVE_ADDRESS 0x04

Servo servo[4][3];

const int servo_pin[4][3] = { {2, 3, 4}, {5, 6, 7}, {8, 9, 10}, {11, 12, 13} };

int number ;

int wart ;

int power = 0 ;

int zmiana = 1 ;

int flagowa_przeszkoda = 0 ;

int mode_lewo_prawo = 0 ;

volatile float polozenie_teraz [4][3] ;

volatile float polozenie_wlasciwe[4][3] ;

float speede [4][3] ;

float predkosc_ruchu ;

float mnoznik_predkosci = 1 ;

float predkosc_obrotu_miejscowego = 4 ;

float predkosc_ruchu1 = 8 ;

float predkosc_ruchu2 = 3 ;

float LightResistant ;

float logR2, T, Tc ;

float resistance ;

float c1 = 1.00924e-03, c2 = 2.378405444e-04, c3 = 2.019202697e-07 ;

[illegible]

```

void setup()
{
    Serial.begin(9600)                ;
    Wire.begin(SLAVE_ADDRESS)        ;
    Wire.onReceive(receiveEvent)      ;
    // Wire.onRequest(requestEvent)   ;
    pinMode (ledpin , OUTPUT        ) ;
    ustawienie(0, x_default - x_offset, y_start + y_step, 40) ;
    ustawienie(1, x_default - x_offset, y_start + y_step, 40) ;
    ustawienie(2, x_default + x_offset, y_start,40)            ;
    ustawienie(3, x_default + x_offset, y_start, 40)          ;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            polozenie_teraz[i][j] = polozenie_wlasciwe[i][j] ;
        }
    }
    FlexiTimer2::set(20, Serwa)        ;
    FlexiTimer2::start()                ;
    zalacz_serwa()                      ;
    while (!Serial)
    {
        ;
    }
    digitalWrite(A4,HIGH)                ;
}

void loop()
{

```

```

int tmp_turn, tmp_leg, tmp_body      ;

if (is_stand())
{
    tmp_turn    =  predkosc_obrotu_miejscowego      ;
    tmp_leg     =  predkosc_ruchu1                  ;
    tmp_body    =  predkosc_ruchu2                  ;
    predkosc_obrotu_miejscowego =  predkosc_ruchu1 = predkosc_ruchu2 = 20 ;
    if (flagowa_przeszkoda < 3)
    {
        krok_w tyl (1)                                ;
        flagowa_przeszkoda++                            ;
    }
else
    {
        if (mode_lewo_prawo)
            skrec_w_prawo (1)                            ;
        else
            skrec_w_lewo (1)                                ;
            mode_lewo_prawo = 1 - mode_lewo_prawo      ;
            flagowa_przeszkoda = 0                        ;
    }
    predkosc_obrotu_miejscowego =  tmp_turn            ;
    predkosc_ruchu1 =  tmp_leg                          ;
    predkosc_ruchu2 =  tmp_body                          ;
}

while (true)
{
    ///////////////////////////////////TEMPERATURA////////////////////////////////////
    odczyt    =  analogRead(A1)                            ;

```

```

resistance = seriesresistor * ((1023.0/odczyt )-1 )      ;
logR2      = log(resistance)                             ;
T          = (1.0 / (c1 + c2*logR2 + c3*logR2*logR2*logR2)) ;
Tc         = T - 273.15                                   ;
Serial.println (Tc)                                       ;

//////////////////////////////////OSWIETLENIE_KAMERY//////////////////////////////////

analogWrite (ledpin,power)      ;
LightResistant = analogRead(A2) ;
if (LightResistant < 30)
{
    power++                      ;
}
if (LightResistant > 120)
{
    power--                      ;
}
if (power>255)
{
    power=255                    ;
}
if (power<0)
{
    power=0                      ;
}

//////////////////////////////////STEROWANIE//////////////////////////////////

if( number == 119)
{
    Serial.println("Do przodu") ;
    krok_naprzod(5)             ;
}

```

```

        number = 0          ;
        false               ;
    }
else if ( number == 115 )
{
    Serial.println("Do tylu")  ;
    krok_w_tyl (5)           ;
    number = 0               ;
    false                   ;
}
else if( number == 100 )
{
    Serial.println("W prawo") ;
    skrec_w_prawo (5)        ;
    number = 0               ;
    false                   ;
}
else if( number == 97 )
{
    Serial.println("W lewo") ;
    skrec_w_lewo (5)         ;
    number = 0               ;
    false                   ;
}
else if( number == 122 )
{
    Serial.println("Z")       ;
    sit ()                   ;
    number = 0               ;

```

```
        false                ;  
    }  
    else if( number == 120 )  
    {  
        Serial.println("X")    ;  
        stand ()              ;  
        number = 0            ;  
        false                 ;  
    }  
    else if( number ==99 )  
    {  
        Serial.println("C")    ;  
        hand(3)                ;  
        number = 0            ;  
        false                 ;  
    }  
    else if( number ==110 )  
    {  
        Serial.println("N")    ;  
        hand_1(3)              ;  
        number = 0            ;  
        false                 ;  
    }  
    else if( number ==98 )  
    {  
        Serial.println("B")    ;  
        right(3)               ;  
        left(3)                ;  
        number = 0            ;
```



```
        false                ;  
    }  
    else if( number ==109 )  
    {  
        Serial.println("M")    ;  
        head_down(10)        ;  
        number = 0            ;  
        false                ;  
    }  
    else if( number ==108 )  
    {  
        Serial.println("L")    ;  
        head_up(10)           ;  
        number = 0            ;  
        false                ;  
    }  
    else if( number ==107 )  
    {  
        Serial.println("K")    ;  
        body_right(10)         ;  
        number = 0            ;  
        false                ;  
    }  
    else if( number ==106 )  
    {  
        Serial.println("J")    ;  
        body_left(10)          ;  
        number = 0            ;  
        false                ;  
    }
```

```

    }
    else if( number ==102 )
    {
        Serial.println("F")    ;
        poczatkowa(1)        ;
        number = 0            ;
        false                  ;
    }
}
}

////////////////////////////////////I2C////////////////////////////////////

void receiveEvent(int howMany)
{
    while (1 < Wire.available())
    {
        char c = Wire.read()  ;
    }
    int x = Wire.read()      ;
    if( x == 119)
    {
        number=119          ;
    }
    else if (x == 115 )
    {
        number = 115        ;
    }
    else if( x ==100 )
    {
        number = 100        ;
    }
}

```

```
    }  
else if( x == 97 )  
{  
    number = 97    ;  
}  
else if( x == 122 )  
{  
    number = 122    ;  
}  
else if( x ==120 )  
{  
    number = 120    ;  
}  
else if( x ==99 )  
{  
    number = 99    ;  
}  
else if( x ==110 )  
{  
    number = 110    ;  
}  
else if( x ==98 )  
{  
    number = 98    ;  
}  
else if( x ==109 )  
{  
    number = 109    ;  
}
```

```
    else if( x ==108 )
    {
        number = 108        ;
    }
    else if( x ==107 )
    {
        number = 107        ;
    }
    else if( x ==106 )
    {
        number = 106        ;
    }
    else if( x ==104 )
    {
        number = 104        ;
    }
    else if( x ==103 )
    {
        number = 103        ;
    }
    else if( x ==102 )
    {
        number = 102        ;
    }
}

//void requestEvent()
//{
//Wire.write("Tc");
//}
```



```

        }
        czekaj()
    }
void sit(void)
{
    predkosc_ruchu = predkosc_siedz_wstan
    for (int leg = 0; leg < 4; leg++)
    {
        ustawienie(leg, USTAW, USTAW, z_boot)
    }
    czekaj()
}
bool is_stand(void)
{
    if (polozenie_teraz[0][2] == z_default)
        return true
    else
        return false
}

void czekaj(void)
{
    for (int i = 0; i < 4; i++)
        czekaj_1(i);
}

void czekaj_1(int leg)
{
    while (1)

```

```

        if (polozenie_teraz[leg][0] == polozenie_wlasciwe[leg][0])
            if (polozenie_teraz[leg][1] == polozenie_wlasciwe[leg][1])
                if (polozenie_teraz[leg][2] == polozenie_wlasciwe[leg][2])
                    break;
    }

void zalacz_serwa(void)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            servo[i][j].attach(servo_pin[i][j])  ;
            delay(100)                            ;
        }
    }
}

void odlacz_serwa(void)
{
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            servo[i][j].detach()                  ;
            delay(100)                             ;
        }
    }
}

```

```

void Serwa(void)
{
    sei() ;
    static float alpha, beta, gamma ;
    for (int i = 0; i < 4; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            if (abs(polozenie_teraz[i][j] - polozenie_wlasciwe[i][j]) >= abs(speede[i][j]))
                polozenie_teraz[i][j] += speede[i][j] ;
            else
                polozenie_teraz[i][j] = polozenie_wlasciwe[i][j] ;
        }

        kartezyjanski_na_biegunowy(alpha, beta, gamma, polozenie_teraz[i][0],
        polozenie_teraz[i][1], polozenie_teraz[i][2]) ;
        biegun_dla_serw(i, alpha, beta, gamma) ;
    }
    licznik++;
}

void kartezyjanski_na_biegunowy(volatile float &alpha, volatile float &beta, volatile float
&gamma, volatile float x, volatile float y, volatile float z)
{
    float v, w ;
    w = (x >= 0 ? 1 : -1) * (sqrt(pow(x, 2) + pow(y, 2))) ;
    v = w - coxa ;
    alpha = atan2(z, v) + acos((pow(femur, 2) - pow(tibia, 2) + pow(v, 2) + pow(z, 2)) / 2 /
femur / sqrt(pow(v, 2) + pow(z, 2))) ;
}

```



```

    beta = acos((pow(femur, 2) + pow(tibia, 2) - pow(v, 2) - pow(z, 2)) / 2 / femur / tibia) ;
    gamma = (w >= 0) ? atan2(y, x) : atan2(-y, -x) ;
    alpha = alpha / pi * 180 ;
    beta = beta / pi * 180 ;
    gamma = gamma / pi * 180 ;
}

void biegun_dla_serw(int leg, float alpha, float beta, float gamma)
{
    if (leg == 0)
    {
        alpha = 90 - alpha ;
        beta = beta ;
        gamma += 90 ;
    }
    else if (leg == 1)
    {
        alpha += 90 ;
        beta = 180 - beta ;
        gamma = 90 - gamma ;
    }
    else if (leg == 2)
    {
        alpha += 90 ;
        beta = 180 - beta ;
        gamma = 90 - gamma ;
    }
    else if (leg == 3)
    {
        alpha = 90 - alpha ;

```



```

    predkosc_ruchu = predkosc_ruchu1 ;

    ustawienie(1, x_default + x_offset, y_start + 2 * y_step, z_up) ;
    czekaj() ;
    ustawienie(1, x_default + x_offset, y_start, z_up) ;
    czekaj() ;
    ustawienie(1, x_default + x_offset, y_start, z_default) ;
    czekaj() ;
}
else
{
    ustawienie(0, x_default + x_offset, y_start, z_up) ;
    czekaj() ;
    ustawienie(0, x_default + x_offset, y_start + 2 * y_step, z_up) ;
    czekaj() ;
    ustawienie(0, x_default + x_offset, y_start + 2 * y_step, z_default) ;
    czekaj() ;

    predkosc_ruchu = predkosc_ruchu2 ;

    ustawienie(0, x_default - x_offset, y_start + y_step, z_default) ;
    ustawienie(1, x_default - x_offset, y_start + y_step, z_default) ;
    ustawienie(2, x_default + x_offset, y_start, z_default) ;
    ustawienie(3, x_default + x_offset, y_start + 2 * y_step, z_default) ;
    czekaj() ;

    predkosc_ruchu = predkosc_ruchu1 ;

```

```

        ustawienie(3, x_default + x_offset, y_start + 2 * y_step, z_up) ;
        czekaj() ;
        ustawienie(3, x_default + x_offset, y_start, z_up) ;
        czekaj() ;
        ustawienie(3, x_default + x_offset, y_start, z_default) ;
        czekaj() ;
    }

}

void krok_w tyl(unsigned int step)
{
    predkosc_ruchu = predkosc_ruchu1;
    while (step-- > 0)
    {
        if (polozenie_teraz[3][1] == y_start)
        {
            ustawienie(3, x_default + x_offset, y_start, z_up) ;
            czekaj() ;
            ustawienie(3, x_default + x_offset, y_start + 2 * y_step, z_up) ;
            czekaj() ;
            ustawienie(3, x_default + x_offset, y_start + 2 * y_step, z_default) ;
            czekaj() ;
            predkosc_ruchu = predkosc_ruchu2 ;
            ustawienie(0, x_default + x_offset, y_start + 2 * y_step, z_default) ;
            ustawienie(1, x_default + x_offset, y_start, z_default) ;
            ustawienie(2, x_default - x_offset, y_start + y_step, z_default) ;
            ustawienie(3, x_default - x_offset, y_start + y_step, z_default) ;
            czekaj() ;
        }
    }
}

```

```

    predkosc_ruchu = predkosc_ruchu1 ;
    ustawienie(0, x_default + x_offset, y_start + 2 * y_step, z_up) ;
    czekaj() ;
    ustawienie(0, x_default + x_offset, y_start, z_up) ;
    czekaj() ;
    ustawienie(0, x_default + x_offset, y_start, z_default) ;
    czekaj() ;
}
else
{
    ustawienie(1, x_default + x_offset, y_start, z_up) ;
    czekaj() ;
    ustawienie(1, x_default + x_offset, y_start + 2 * y_step, z_up) ;
    czekaj() ;
    ustawienie(1, x_default + x_offset, y_start + 2 * y_step, z_default) ;
    czekaj() ;
    predkosc_ruchu = predkosc_ruchu2 ;
    ustawienie(0, x_default - x_offset, y_start + y_step, z_default) ;
    ustawienie(1, x_default - x_offset, y_start + y_step, z_default) ;
    ustawienie(2, x_default + x_offset, y_start + 2 * y_step, z_default) ;
    ustawienie(3, x_default + x_offset, y_start, z_default) ;
    czekaj() ;
    predkosc_ruchu = predkosc_ruchu1 ;
    ustawienie(2, x_default + x_offset, y_start + 2 * y_step, z_up) ;
    czekaj() ;
    ustawienie(2, x_default + x_offset, y_start, z_up) ;
    czekaj() ;
    ustawienie(2, x_default + x_offset, y_start, z_default) ;
    czekaj() ;
}

```

```

    }
}
}
void skrec_w_prawo(unsigned int step)
{
    predkosc_ruchu = predkosc_obrotu_miejscowego;
    while (step-- > 0)
    {
        if (polozenie_teraz[2][1] == y_start)
        {
            ustawienie(2, x_default + x_offset, y_start, z_up) ;
            czekaj() ;
            ustawienie(0, turn_x0 - x_offset, turn_y0, z_default) ;
            ustawienie(1, turn_x1 - x_offset, turn_y1, z_default) ;
            ustawienie(2, turn_x0 + x_offset, turn_y0, z_up) ;
            ustawienie(3, turn_x1 + x_offset, turn_y1, z_default) ;
            czekaj() ;
            ustawienie(2, turn_x0 + x_offset, turn_y0, z_default) ;
            czekaj() ;
            ustawienie(0, turn_x0 + x_offset, turn_y0, z_default) ;
            ustawienie(1, turn_x1 + x_offset, turn_y1, z_default) ;
            ustawienie(2, turn_x0 - x_offset, turn_y0, z_default) ;
            ustawienie(3, turn_x1 - x_offset, turn_y1, z_default) ;
            czekaj() ;
            ustawienie(0, turn_x0 + x_offset, turn_y0, z_up) ;
            czekaj() ;
            ustawienie(0, x_default + x_offset, y_start, z_up) ;
            ustawienie(1, x_default + x_offset, y_start, z_default) ;
            ustawienie(2, x_default - x_offset, y_start + y_step, z_default) ;

```

```

ustawienie(3, x_default - x_offset, y_start + y_step, z_default) ;
czekaj() ;
ustawienie(0, x_default + x_offset, y_start, z_default) ;
czekaj() ;
}
else
{
ustawienie(1, x_default + x_offset, y_start, z_up) ;
czekaj() ;
ustawienie(0, turn_x1 + x_offset, turn_y1, z_default) ;
ustawienie(1, turn_x0 + x_offset, turn_y0, z_up) ;
ustawienie(2, turn_x1 - x_offset, turn_y1, z_default) ;
ustawienie(3, turn_x0 - x_offset, turn_y0, z_default) ;
czekaj() ;
ustawienie(1, turn_x0 + x_offset, turn_y0, z_default) ;
czekaj() ;
ustawienie(0, turn_x1 - x_offset, turn_y1, z_default) ;
ustawienie(1, turn_x0 - x_offset, turn_y0, z_default) ;
ustawienie(2, turn_x1 + x_offset, turn_y1, z_default) ;
ustawienie(3, turn_x0 + x_offset, turn_y0, z_default) ;
czekaj() ;
ustawienie(3, turn_x0 + x_offset, turn_y0, z_up) ;
czekaj() ;
ustawienie(0, x_default - x_offset, y_start + y_step, z_default) ;
ustawienie(1, x_default - x_offset, y_start + y_step, z_default) ;
ustawienie(2, x_default + x_offset, y_start, z_default) ;
ustawienie(3, x_default + x_offset, y_start, z_up) ;
czekaj() ;
ustawienie(3, x_default + x_offset, y_start, z_default) ;

```

```

        czekaj()                                ;
    }
}
}

void skrec_w_lewo(unsigned int step)
{
    predkosc_ruchu = predkosc_obrotu_miejscowego ;
    while (step-- > 0)
    {
        if (polozenie_teraz[3][1] == y_start)
        {
            ustawienie(3, x_default + x_offset, y_start, z_up) ;
            czekaj() ;
            ustawienie(0, turn_x1 - x_offset, turn_y1, z_default) ;
            ustawienie(1, turn_x0 - x_offset, turn_y0, z_default) ;
            ustawienie(2, turn_x1 + x_offset, turn_y1, z_default) ;
            ustawienie(3, turn_x0 + x_offset, turn_y0, z_up) ;
            czekaj() ;
            ustawienie(3, turn_x0 + x_offset, turn_y0, z_default) ;
            czekaj() ;
            ustawienie(0, turn_x1 + x_offset, turn_y1, z_default) ;
            ustawienie(1, turn_x0 + x_offset, turn_y0, z_default) ;
            ustawienie(2, turn_x1 - x_offset, turn_y1, z_default) ;
            ustawienie(3, turn_x0 - x_offset, turn_y0, z_default) ;
            czekaj() ;
            ustawienie(1, turn_x0 + x_offset, turn_y0, z_up) ;
            czekaj() ;
            ustawienie(0, x_default + x_offset, y_start, z_default) ;
            ustawienie(1, x_default + x_offset, y_start, z_up) ;

```



```

ustawienie(2, x_default - x_offset, y_start + y_step, z_default) ;
ustawienie(3, x_default - x_offset, y_start + y_step, z_default) ;
czekaj() ;
ustawienie(1, x_default + x_offset, y_start, z_default) ;
czekaj() ;
}
else
{
ustawienie(0, x_default + x_offset, y_start, z_up) ;
czekaj() ;
ustawienie(0, turn_x0 + x_offset, turn_y0, z_up) ;
ustawienie(1, turn_x1 + x_offset, turn_y1, z_default) ;
ustawienie(2, turn_x0 - x_offset, turn_y0, z_default) ;
ustawienie(3, turn_x1 - x_offset, turn_y1, z_default) ;
czekaj() ;
ustawienie(0, turn_x0 + x_offset, turn_y0, z_default) ;
czekaj() ;
ustawienie(0, turn_x0 - x_offset, turn_y0, z_default) ;
ustawienie(1, turn_x1 - x_offset, turn_y1, z_default) ;
ustawienie(2, turn_x0 + x_offset, turn_y0, z_default) ;
ustawienie(3, turn_x1 + x_offset, turn_y1, z_default) ;
czekaj() ;
ustawienie(2, turn_x0 + x_offset, turn_y0, z_up) ;
czekaj() ;
ustawienie(0, x_default - x_offset, y_start + y_step, z_default) ;
ustawienie(1, x_default - x_offset, y_start + y_step, z_default) ;
ustawienie(2, x_default + x_offset, y_start, z_up) ;
ustawienie(3, x_default + x_offset, y_start, z_default) ;
czekaj() ;

```

```

        ustawienie(2, x_default + x_offset, y_start, z_default)
        ;

        czekaj()
        ;

    }

}

}

void hand(int i)
{
    float x_tmp
        ;

    float y_tmp
        ;

    float z_tmp
        ;

    predkosc_ruchu = 1
        ;

    if (polozenie_teraz[3][1] == y_start)
    {
        body_right(15)
            ;

        x_tmp = polozenie_teraz[2][0]
            ;

        y_tmp = polozenie_teraz[2][1]
            ;

        z_tmp = polozenie_teraz[2][2]
            ;

        predkosc_ruchu = predkosc_ruchu2
            ;

        for (int j = 0; j < i; j++)
        {
            ustawienie(2, turn_x1, turn_y1, 50)
                ;

            czekaj()
                ;

            ustawienie(2, turn_x0, turn_y0, 50)
                ;

            czekaj()
                ;

        }

        ustawienie(2, x_tmp, y_tmp, z_tmp)
            ;

        czekaj()
            ;

        predkosc_ruchu = 1
            ;

        body_left(15)
            ;
    }
}

```

```

    }
else
{
    body_left(15)                ;
    x_tmp = polozenie_teraz[0][0] ;
    y_tmp = polozenie_teraz[0][1] ;
    z_tmp = polozenie_teraz[0][2] ;
    predkosc_ruchu = predkosc_ruchu2 ;
    for (int j = 0; j < i; j++)
    {
        ustawienie(0, turn_x1, turn_y1, 50) ;
        czekaj() ;
        ustawienie(0, turn_x0, turn_y0, 50) ;
        czekaj() ;
    }
    ustawienie(0, x_tmp, y_tmp, z_tmp) ;
    czekaj() ;
    predkosc_ruchu = 1 ;
    body_right(15) ;
}
}

void hand_1(int i)
{
    float x_tmp ;
    float y_tmp ;
    float z_tmp ;
    predkosc_ruchu = 1 ;
    if (polozenie_teraz[3][1] == y_start)
    {

```

```

body_right(15)                                ;
x_tmp = polozenie_teraz[2][0]                 ;
y_tmp = polozenie_teraz[2][1]                 ;
z_tmp = polozenie_teraz[2][2]                 ;
predkosc_ruchu = predkosc_ruchu2             ;
for (int j = 0; j < i; j++)
{
    ustawienie(2, x_default - 30, y_start + 2 * y_step, 55) ;
    czekaj()                                           ;
    ustawienie(2, x_default - 30, y_start + 2 * y_step, 10) ;
    czekaj()                                           ;
}
ustawienie(2, x_tmp, y_tmp, z_tmp) ;
czekaj()                                           ;
predkosc_ruchu = 1                                ;
body_left(15)                                       ;
}
else
{
    body_left(15)                                       ;
    x_tmp = polozenie_teraz[0][0]                 ;
    y_tmp = polozenie_teraz[0][1]                 ;
    z_tmp = polozenie_teraz[0][2]                 ;
    predkosc_ruchu = predkosc_ruchu2             ;
    for (int j = 0; j < i; j++)
    {
        ustawienie(0, x_default - 30, y_start + 2 * y_step, 55) ;
        czekaj()                                           ;
        ustawienie(0, x_default - 30, y_start + 2 * y_step, 10) ;
    }
}

```

```

        czekaj() ;
    }
    ustawienie(0, x_tmp, y_tmp, z_tmp) ;
    czekaj() ;
    predkosc_ruchu = 1 ;
    body_right(15) ;
}
}
void body_left(int i)
{
    ustawienie(0, polozenie_teraz[0][0] + i, USTAW, USTAW) ;
    ustawienie(1, polozenie_teraz[1][0] + i, USTAW, USTAW) ;
    ustawienie(2, polozenie_teraz[2][0] - i, USTAW, USTAW) ;
    ustawienie(3, polozenie_teraz[3][0] - i, USTAW, USTAW) ;
    czekaj() ;
}
void body_right(int i)
{
    ustawienie(0, polozenie_teraz[0][0] - i, USTAW, USTAW) ;
    ustawienie(1, polozenie_teraz[1][0] - i, USTAW, USTAW) ;
    ustawienie(2, polozenie_teraz[2][0] + i, USTAW, USTAW) ;
    ustawienie(3, polozenie_teraz[3][0] + i, USTAW, USTAW) ;
    czekaj() ;
}

```

```

void head_up(int i)
{
    ustawienie(0, USTAW, USTAW, polozenie_teraz[0][2] - i)    ;
    ustawienie(1, USTAW, USTAW, polozenie_teraz[1][2] + i)    ;
    ustawienie(2, USTAW, USTAW, polozenie_teraz[2][2] - i)    ;
    ustawienie(3, USTAW, USTAW, polozenie_teraz[3][2] + i)    ;
    czekaj()                                                    ;
}

```

```

void head_down(int i)
{
    ustawienie(0, USTAW, USTAW, polozenie_teraz[0][2] + i)    ;
    ustawienie(1, USTAW, USTAW, polozenie_teraz[1][2] - i)    ;
    ustawienie(2, USTAW, USTAW, polozenie_teraz[2][2] + i)    ;
    ustawienie(3, USTAW, USTAW, polozenie_teraz[3][2] - i)    ;
    czekaj()                                                    ;
}

```

```

void right(int i)
{
    float x_tmp                                                ;
    float y_tmp                                                ;
    float z_tmp                                                ;
    predkosc_ruchu = 1                                         ;
    body_right(15)                                              ;
    x_tmp = polozenie_teraz[2][0]                               ;
    y_tmp = polozenie_teraz[2][1]                               ;
    z_tmp = polozenie_teraz[2][2]                               ;
    predkosc_ruchu = predkosc_ruchu2                           ;
}

```

```
}
```

```
void left(int i)
```

```
{
```

```
    float x_tmp          ;
```

```
    float y_tmp          ;
```

```
    float z_tmp          ;
```

```
    predkosc_ruchu = 1    ;
```

```
        body_left(15)    ;
```

```
        x_tmp = polozenie_teraz[2][0]    ;
```

```
        y_tmp = polozenie_teraz[2][1]    ;
```

```
        z_tmp = polozenie_teraz[2][2]    ;
```

```
        predkosc_ruchu = predkosc_ruchu2 ;
```

```
}
```

```
void poczatkowa(int i )
```

```
{
```

```
    ustawienie(0, x_default - x_offset, y_start + y_step, 40) ;
```

```
    ustawienie(1, x_default - x_offset, y_start + y_step, 40) ;
```

```
    ustawienie(2, x_default + x_offset, y_start,40)          ;
```

```
    ustawienie(3, x_default + x_offset, y_start, 40)          ;
```

```
}
```