Universitatea POLITEHNICA din București
Facultatea de Inginerie Industrială și Robotică

ROBOTICĂ 1 – TEME DE CASĂ

Student: BUZDUGAN Andrei

Coordonator: Ileana DUGĂEȘESCU

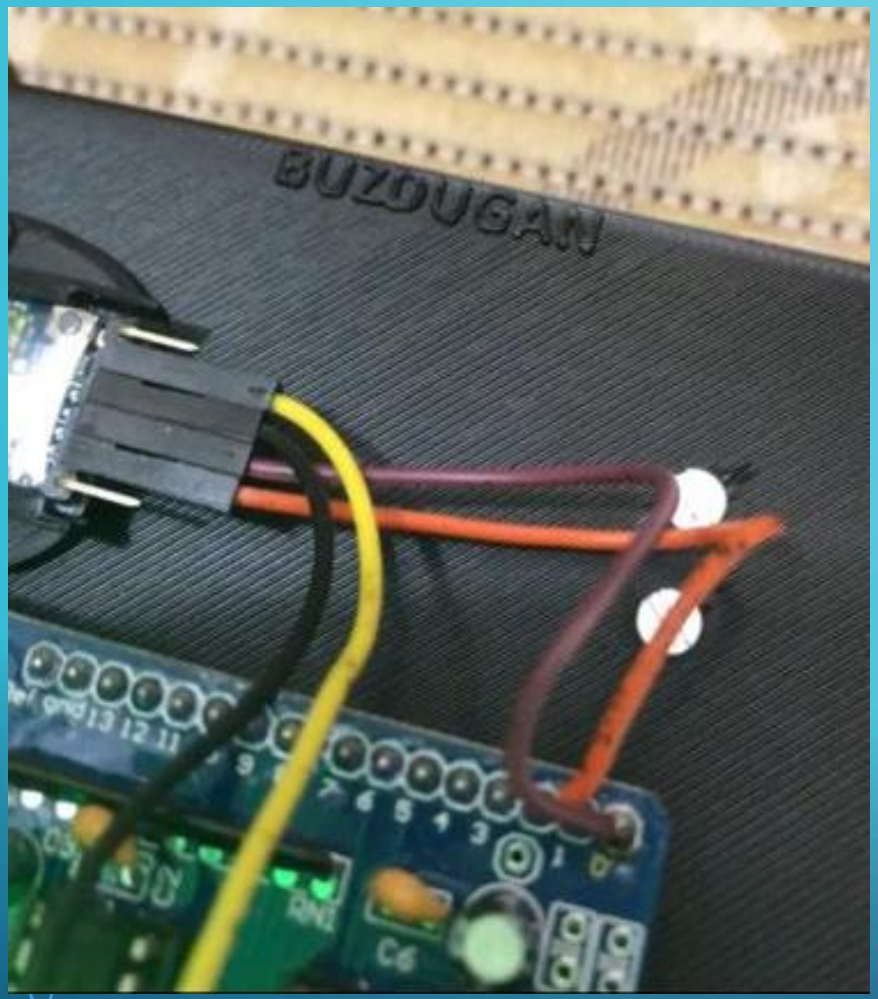
Imprimanta 3D

Am asamblată și programat imprimanta pe care mi-am cumparat-o special pentru a realiza componentele necesare pentru construirea roboților.


Ender


Ender
Ender
Ender

La aceste teme de casa pe care le-am realizat, am învățat să folosesc programul de modelare 3D SOLIDWORKS, cu ajutorul căruia am realizat de la zero unele piese și am adăugat pe fiecare robot numele meu “BUZDUGAN” .

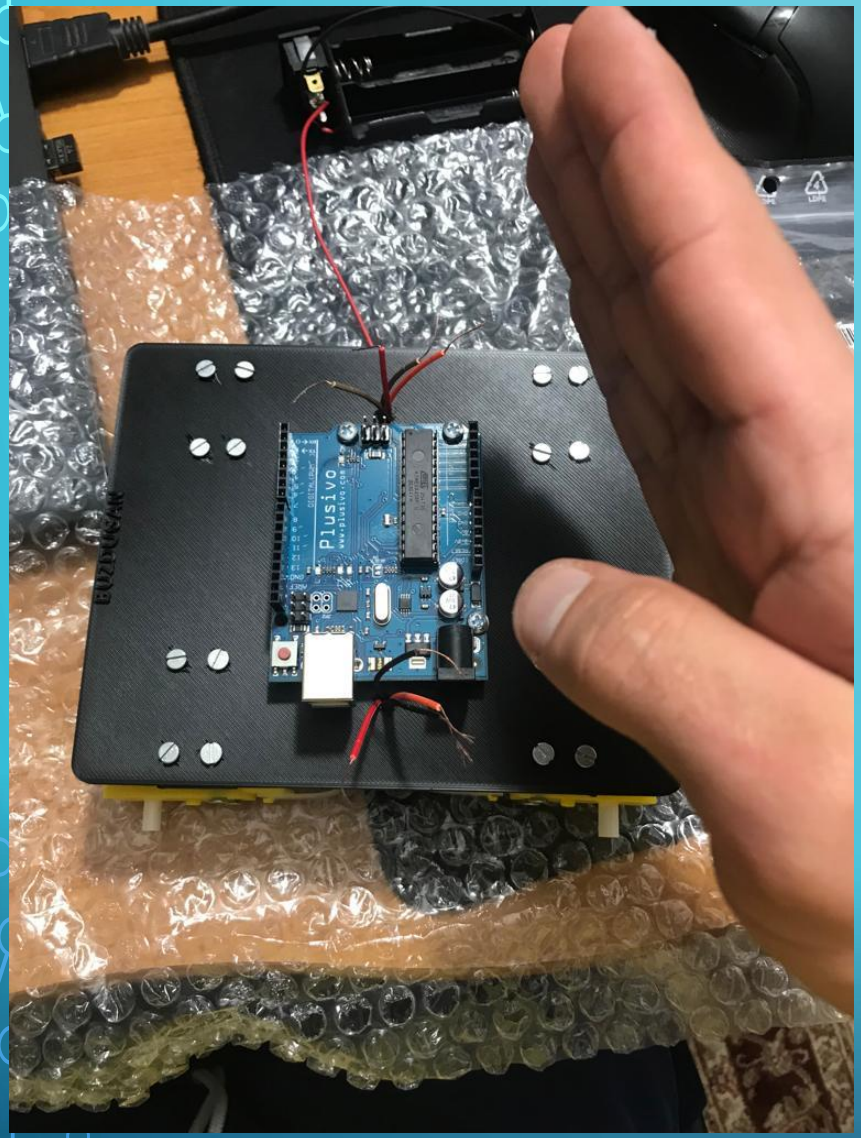


Iar pe lângă cele de mai sus, a trebuit sa învăț să lipesc cu pistolul de lipit niște fire, ba chiar am reușit să lipesc niște pini pe o plăcuță de expansiune.

CODUL: Pentru controlul mașinuței prin bluetooth cu ajutorul unei aplicații Android.

Robot Car


```
#include <AFMotor.h>

AF_DCMotor motor1(1, MOTOR12_1KHZ);
AF_DCMotor motor2(2, MOTOR12_1KHZ);
AF_DCMotor motor3(3, MOTOR34_1KHZ);
AF_DCMotor motor4(4, MOTOR34_1KHZ);

char command;

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

void loop(){
 if(Serial.available() > 0){
  command = Serial.read();

  motor1.setSpeed(0);
  motor1.run(RELEASE);
  motor2.setSpeed(0);
  motor2.run(RELEASE);
  motor3.setSpeed(0);
  motor3.run(RELEASE);
  motor4.setSpeed(0);
  motor4.run(RELEASE);

  switch(command){
  case 'F':
   motor1.setSpeed(255);
   motor1.run(FORWARD);
   motor2.setSpeed(255);
   motor2.run(FORWARD);
   motor3.setSpeed(255);
   motor3.run(FORWARD);
   motor4.setSpeed(255);
   motor4.run(FORWARD);
   break;
case 'B':// Back
   motor1.setSpeed(255);
   motor1.run(BACKWARD);
   motor2.setSpeed(255);
   motor2.run(BACKWARD);
   motor3.setSpeed(255);
   motor3.run(BACKWARD);
   motor4.setSpeed(255);
   motor4.run(BACKWARD);
   break;
  case 'L':// Left
   motor1.setSpeed(255);
   motor1.run(BACKWARD);
   motor2.setSpeed(255);
   motor2.run(BACKWARD);
   motor3.setSpeed(255);
   motor3.run(FORWARD);
   motor4.setSpeed(255);
   motor4.run(FORWARD);
   break;
  case 'R':// Right
   motor1.setSpeed(255);
   motor1.run(FORWARD);
   motor2.setSpeed(255);
   motor2.run(FORWARD);
   motor3.setSpeed(255);
   motor3.run(BACKWARD);
   motor4.setSpeed(255);
   motor4.run(BACKWARD);
   break;
  }
 }
}
```


Plusivo


BUZDUGAN

Robot Boy

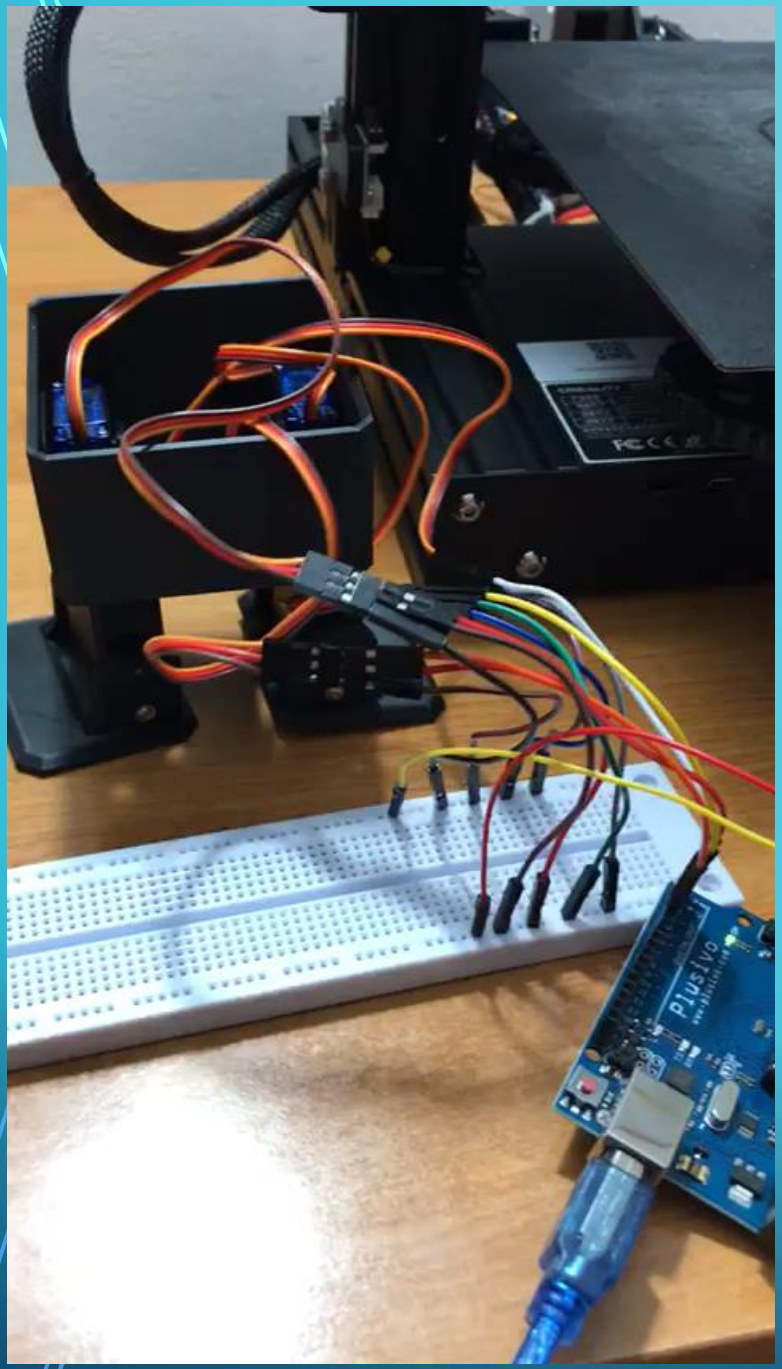
CODUL: Pentru dans.

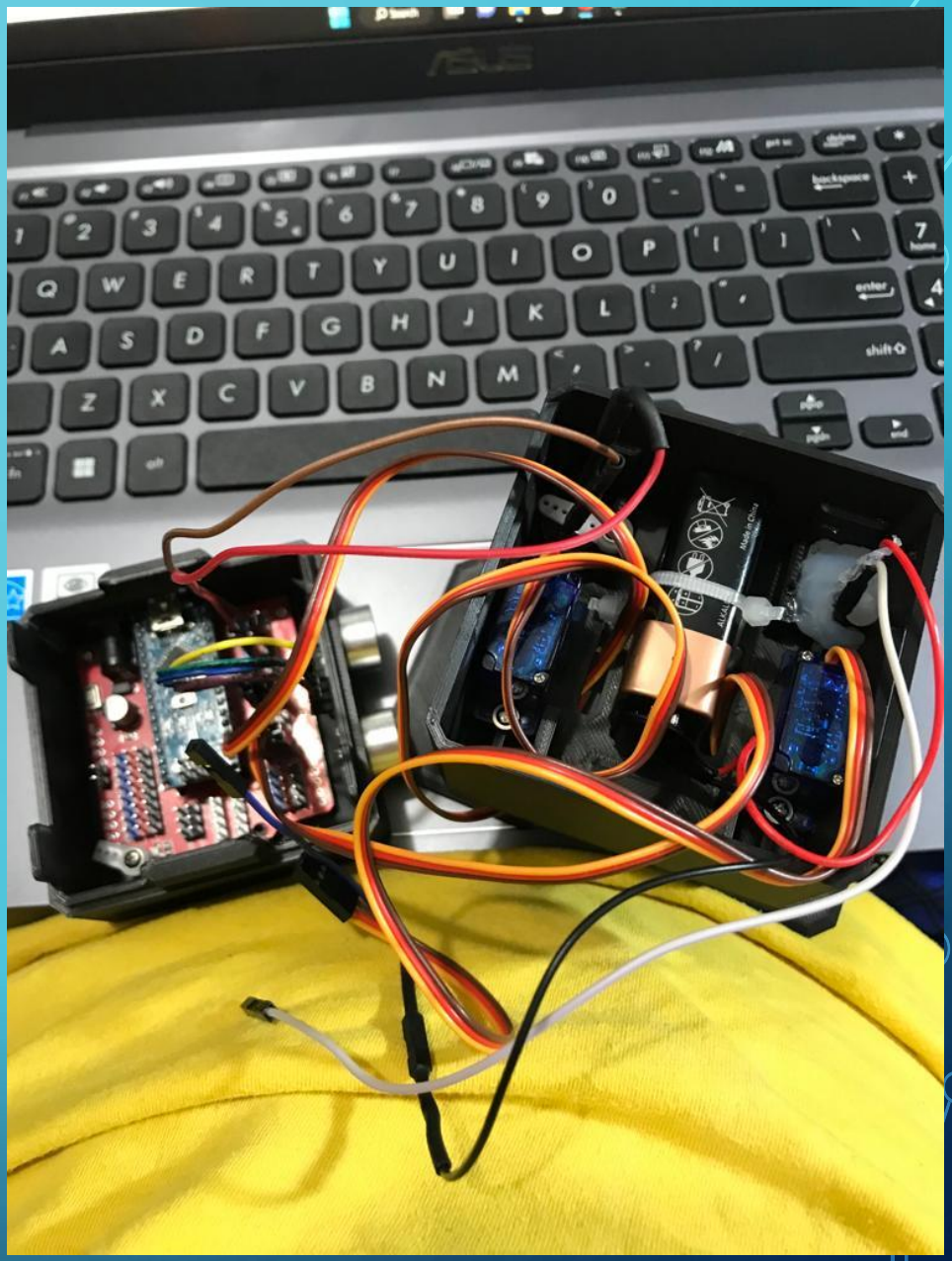
```
#include "VarSpeedServo.h"
#include <NewPing.h>
#include <Servo.h>

VarSpeedServo RU;
VarSpeedServo RL;
VarSpeedServo LU;
VarSpeedServo LL;

NewPing sonar(4,3,200);

const int vel = 40, vel_Back = 15;
const int delay_Forward = 300, delay_Back = 750;


int vel_Dance1 = 30,   vel_Dance2 = 25,   vel_Dance3 = 40;
int delay_Dance1 = 300, delay_Dance2 = 750, delay_Dance3 = 200;

int vel_Dance4 = 40,   vel_Dance5 = 40,   vel_Dance6 = 30;
int delay_Dance4 = 400, delay_Dance5 = 400, delay_Dance6 = 500;

const int array_cal[4] = {90,90,90,90};
int RU_Degree = 0, LU_Degree = array_cal[2] + 5;

const int num_dance1 = 10;
const int array_dance1[num_dance1][4] =
{

   {0,-20,0,0},
   {0,-40,0,20},
   {0,-20,0,40},
   {0,0,0,20},
   {0,0,0,0},


   {0,0,0,20},
   {0,-20,0,40},
   {0,-40,0,20},
   {0,-20,0,0},
   {0,0,0,0},
};

const int num_dance2 = 32;
const int array_dance2[num_dance2][4] =
{

   {20,0,40,0},
   {20,-30,40,-30},
   {20,-30,10,-30},
   {20,-30,40,-30},
   {20,-30,10,-30},

   {20,-30,40,-30},
   {20,0,40,-30},
   {20,80,40,-30},
   {20,0,40,-30},
   {20,-80,40,-30},
   {20,0,40,-30},
   {20,80,40,-30},
   {20,0,40,-30},
   {20,-30,40,-30},
   {20,0,40,0},
   {0,0,0,0},


   {-40,0,-20,0},
   {-40,40,-20,30},
   {-20,40,-20,30},
   {-40,40,-20,30},
   {-20,40,-20,30},

   {-40,40,-20,30},
   {-40,40,-20,0},
   {-40,40,-20,-80},
   {-40,40,-20,0},
   {-40,40,-20,80},
   {-40,40,-20,0},
   {-40,40,-20,-80},
   {-40,40,-20,0},
   {-40,40,-20,30},
   {-40,0,-20,0},
   {0,0,0,0},
};

const int num_dance3 = 8;
const int array_dance3[num_dance3][4] =
{
   {0,-40,0,0},
   {20,-30,20,20},
   {40,0,40,30},
   {0,0,0,-10},
   {-20,-20,-20,30},
   {-40,-30,-40,0},

   {0,-40,0,0},
   {0,0,0,0},
};

const int num_dance4 = 20;
const int array_dance4[num_dance4][4] =
{
   {0,-20,0,20},
   {0,0,0,0},
   {0,-20,0,20},
   {0,0,0,0},
   {0,-20,0,20},
   {0,0,0,0},
   {0,-20,0,20},
   {0,0,0,0},

   {0,-50,0,50},
   {0,0,0,0},
   {0,-50,0,50},
   {0,0,0,0},
   {0,-50,0,50},
   {0,0,0,0},
   {0,-50,0,50},
   {0,0,0,0},

   {0,-40,0,40},
   {0,-50,0,50},
   {0,-60,0,60},
   {0,0,0,0,},
};

const int num_dance5 = 17;
const int array_dance5[num_dance5][4] =
{
   {35,0,15,0},
   {35,30,15,30},
   {-35,30,15,30},
   {-20,0,15,0},
   {0,0,0,0},

   {0,-40,0,40},
   {-30,-40,-20,40},
   {0,-40,0,40},
   {20,-40,30,40},


   {0,-40,0,40},
   {20,-40,-20,40},
   {20,-20,-20,20},
   {20,0,-20,0},
   {-20,-10,20,10},
   {-10,-30,10,30},

   {0,-40,0,40},
   {0,0,0,0},
};

const int num_dance6 = 32;
const int array_dance6[num_dance6][4] =
{
   {0,-40,0,-20},
   {25,-40,18,-20},
   {25,0,18,0},
   {0,20,0,40},
   {-18,20,-25,40},
   {-18,0,-25,0},

   {0,-40,0,-20},
   {25,-40,18,-20},
   {0,0,0,0},
   {0,20,0,40},
   {-18,20,-25,40},
   {0,0,0,0},

   {0,-40,0,-20},
   {-25,-40,-18,-20},
   {-25,0,-18,0},
   {0,20,0,40},
   {18,20,25,40},
   {18,0,25,0},

   {0,-40,0,-20},
   {30,-40,30,-20},
   {30,0,30,0},
   {0,20,0,40},
   {-30,20,-30,40},
   {-30,0,-30,0},

   {0,-40,0,-20},
   {-30,-40,-30,-20},
   {-30,0,-30,0},
   {0,20,0,40},
   {30,20,30,40},
   {30,0,30,0},

   {15,0,15,0},
   {0,0,0,0},
};
```

```
#define RUN

void Servo_Init()
{
  RU.attach(9);   /
  RL.attach(10);
  LU.attach(11);
  LL.attach(12);
}

void Adjust()
{
  for(RU_Degree = array_cal[0] - 5; RU_Degree <= array_cal[0]; RU_Degree += 1) {
    RU.write(RU_Degree);
    LU.write(LU_Degree--);
    delay(15);
  }
}

bool TooClose()
{
  int tooclose = 0;
  for(int a=0; a<5; a++) {
    delay(50);
    int din = sonar.ping_in();
    if (din < 7 && din > 0) tooclose++;
  }
  if (tooclose < 5) return 1;
  return 0;
}

void Slide_2_Left(int times)
{
  for(int time1 = 0; time1 < times; time1++) {
    for(int z=0; z<5; z++) {
      RU.slowmove (array_cal[0] + array_dance1[z][0] , vel_Dance1);
      RL.slowmove (array_cal[1] + array_dance1[z][1] , vel_Dance1);
      LU.slowmove (array_cal[2] + array_dance1[z][2] , vel_Dance1);
      LL.slowmove (array_cal[3] + array_dance1[z][3] , vel_Dance1);
      delay(delay_Dance1);
    }
  }
}

void Slide_2_Right(int times)
{
  for(int time1 = 0; time1 < times; time1++) {
    for(int z=5; z<10; z++) {
      RU.slowmove (array_cal[0] + array_dance1[z][0] , vel_Dance1);
      RL.slowmove (array_cal[1] + array_dance1[z][1] , vel_Dance1);
      LU.slowmove (array_cal[2] + array_dance1[z][2] , vel_Dance1);
      LL.slowmove (array_cal[3] + array_dance1[z][3] , vel_Dance1);
      delay(delay_Dance1);
    }
  }
}

void Left_Foot_Support()
{
  for(int z=0; z<16; z++) {
    if ( z > 5 && z < 14) {
      vel_Dance2 = 50;
      delay_Dance2 = 200;
    }
    else {
    vel_Dance2 = 25;
    delay_Dance2 = 750;
    }

    RU.slowmove (array_cal[0] + array_dance2[z][0] , vel_Dance2);
    RL.slowmove (array_cal[1] + array_dance2[z][1] , vel_Dance2);
    LU.slowmove (array_cal[2] + array_dance2[z][2] , vel_Dance2);
    LL.slowmove (array_cal[3] + array_dance2[z][3] , vel_Dance2);
    delay(delay_Dance2);
  }
}

void Right_Foot_Support()
{
  for(int z=16; z<32; z++) {
    if ( z > 21 && z < 30) {
      vel_Dance2 = 50;
      delay_Dance2 = 200;
    }
    else {
    vel_Dance2 = 25;
    delay_Dance2 = 750;
    }

    RU.slowmove (array_cal[0] + array_dance2[z][0] , vel_Dance2);
    RL.slowmove (array_cal[1] + array_dance2[z][1] , vel_Dance2);
    LU.slowmove (array_cal[2] + array_dance2[z][2] , vel_Dance2);
    LL.slowmove (array_cal[3] + array_dance2[z][3] , vel_Dance2);
    delay(delay_Dance2);
  }
}

void Dancing1_2()
{
  Slide_2_Left(2);
  Left_Foot_Support();

  Slide_2_Right(2);
  Right_Foot_Support();
}
```

```
void Dancing3(int Times = 1, int Vel = 40, int Delay = 250, int low = 0, int high = 0)
{
  for(int time3 = 0; time3 < Times; time3++) {
    for(int z=0; z<6; z++) {
      if ( time3 > 1 && time3 < 4) {
      vel_Dance3 = Vel;
      delay_Dance3 = Delay;
      }
      else {
      vel_Dance3 = 40;
      delay_Dance3 = 200;
      }

      RU.slowmove (array_cal[0] + array_dance3[z][0] , vel_Dance3);
      RL.slowmove (array_cal[1] + array_dance3[z][1] , vel_Dance3);
      LU.slowmove (array_cal[2] + array_dance3[z][2] , vel_Dance3);
      LL.slowmove (array_cal[3] + array_dance3[z][3] , vel_Dance3);
      delay(delay_Dance3);
    }
  }
  for(int z=6; z<8; z++) {
      RU.slowmove (array_cal[0] + array_dance3[z][0] , vel_Dance3);
      RL.slowmove (array_cal[1] + array_dance3[z][1] , vel_Dance3);
      LU.slowmove (array_cal[2] + array_dance3[z][2] , vel_Dance3);
      LL.slowmove (array_cal[3] + array_dance3[z][3] , vel_Dance3);
      delay(delay_Dance3);
    }
}

void Dancing4()
{
  for(int z=0; z<num_dance4; z++) {
    if ( z > 17) {
      vel_Dance4 = 10;
      delay_Dance4 = 1500;
      }
    else {
      vel_Dance4 = 40;
      delay_Dance4 = 400;
      }

    RU.slowmove (array_cal[0] + array_dance4[z][0] , vel_Dance4);
    RL.slowmove (array_cal[1] + array_dance4[z][1] , vel_Dance4);
    LU.slowmove (array_cal[2] + array_dance4[z][2] , vel_Dance4);
    LL.slowmove (array_cal[3] + array_dance4[z][3] , vel_Dance4);
    delay(delay_Dance4);
  }
}

void Dancing5()
{
  for(int x = 0; x < 3; x++) {
    for(int z=0; z<5; z++) {
      RU.slowmove (array_cal[0] + array_dance5[z][0] , vel_Dance5);
      RL.slowmove (array_cal[1] + array_dance5[z][1] , vel_Dance5);
      LU.slowmove (array_cal[2] + array_dance5[z][2] , vel_Dance5);
      LL.slowmove (array_cal[3] + array_dance5[z][3] , vel_Dance5);
      delay(delay_Dance5);
    }
  }

  for(int x = 0; x < 2; x++) {
    for(int z=5; z<9; z++) {
      RU.slowmove (array_cal[0] + array_dance5[z][0] , 30);
      RL.slowmove (array_cal[1] + array_dance5[z][1] , 30);
      LU.slowmove (array_cal[2] + array_dance5[z][2] , 30);
      LL.slowmove (array_cal[3] + array_dance5[z][3] , 30);
      delay(550);
    }
  }

  for(int x = 0; x < 3; x++) {
    for(int z=9; z<15; z++) {
      RU.slowmove (array_cal[0] + array_dance5[z][0] , vel_Dance5);
      RL.slowmove (array_cal[1] + array_dance5[z][1] , vel_Dance5);
      LU.slowmove (array_cal[2] + array_dance5[z][2] , vel_Dance5);
      LL.slowmove (array_cal[3] + array_dance5[z][3] , vel_Dance5);
      delay(300);
    }
  }
  for(int z=15; z<17; z++) {
      RU.slowmove (array_cal[0] + array_dance5[z][0] , 10);
      RL.slowmove (array_cal[1] + array_dance5[z][1] , 10);
      LU.slowmove (array_cal[2] + array_dance5[z][2] , 10);
      LL.slowmove (array_cal[3] + array_dance5[z][3] , 10);
      delay(1500);
    }
}

void Dancing6()
{
  const int array_cal_0 = array_cal[0] + 10 , array_cal_2 = array_cal[2] - 10;

  for(int x = 0; x < 3; x++) {
    for(int z=0; z<6; z++) {
      RU.slowmove (array_cal_0 + array_dance6[z][0] , vel_Dance6);
      RL.slowmove (array_cal[1] + array_dance6[z][1] , vel_Dance6);
      LU.slowmove (array_cal_2 + array_dance6[z][2] , vel_Dance6);
      LL.slowmove (array_cal[3] + array_dance6[z][3] , vel_Dance6);
      delay(delay_Dance6);
    }
  }
```

```
for(int x = 0; x < 3; x++) {
    for(int z=6; z<12; z++) {
      RU.slowmove (array_cal_0 + array_dance6[z][0] , 40);
      RL.slowmove (array_cal[1] + array_dance6[z][1] , 40);
      LU.slowmove (array_cal_2 + array_dance6[z][2] , 40);
      LL.slowmove (array_cal[3] + array_dance6[z][3] , 40);
      delay(400);
    }
  }

  for(int x = 0; x < 3; x++) {
    for(int z=12; z<18; z++) {
      RU.slowmove (array_cal_0 + array_dance6[z][0] , vel_Dance6);
      RL.slowmove (array_cal[1] + array_dance6[z][1] , vel_Dance6);
      LU.slowmove (array_cal_2 + array_dance6[z][2] , vel_Dance6);
      LL.slowmove (array_cal[3] + array_dance6[z][3] , vel_Dance6);
      delay(delay_Dance6);
    }
  }
  for(int x = 0; x < 3; x++) {
    for(int z=18; z<24; z++) {
      RU.slowmove (array_cal[0] + array_dance6[z][0] , vel_Dance6);
      RL.slowmove (array_cal[1] + array_dance6[z][1] , vel_Dance6);
      LU.slowmove (array_cal[2] + array_dance6[z][2] , vel_Dance6);
      LL.slowmove (array_cal[3] + array_dance6[z][3] , vel_Dance6);
      delay(delay_Dance6);
    }
  }

  for(int x = 0; x < 3; x++) {
    for(int z=24; z<30; z++) {
      RU.slowmove (array_cal[0] + array_dance6[z][0] , vel_Dance6);
      RL.slowmove (array_cal[1] + array_dance6[z][1] , vel_Dance6);
      LU.slowmove (array_cal[2] + array_dance6[z][2] , vel_Dance6);
      LL.slowmove (array_cal[3] + array_dance6[z][3] , vel_Dance6);
      delay(delay_Dance6);
    }
  }
  for(int z=30; z<32; z++) {
      RU.slowmove (array_cal[0] + array_dance5[z][0] , 10);
      RL.slowmove (array_cal[1] + array_dance5[z][1] , 10);
      LU.slowmove (array_cal[2] + array_dance5[z][2] , 10);
      LL.slowmove (array_cal[3] + array_dance5[z][3] , 10);
      delay(1500);
    }
}

void setup()
{
#ifdef INSTALL
  Servo_Init();

  RU.slowmove (90 , vel);
  RL.slowmove (90 , vel);
  LU.slowmove (90 , vel);
  LL.slowmove (90 , vel);
  while(1);
#endif

#ifdef CALIBRATION
  Servo_Init();
  Adjust();

  RL.slowmove (array_cal[1] , vel);
  LL.slowmove (array_cal[3] , vel);
  delay(2000);
  while(1);
#endif

#ifdef RUN
  Servo_Init();
  Adjust();

  RL.slowmove (array_cal[1] , vel);
  LL.slowmove (array_cal[3] , vel);
  delay(2000);
#endif
}

void loop()
{
  Dancing1_2();
  delay(500);
  Dancing3(5,20,400);
  delay(500);
  Dancing4();
  delay(500);
  Dancing5();
  delay(500);
  Dancing6();
  delay(500);
  while(1);

}
```






BUZOUGAN

CODUL: Pentru evitarea obstacolelor cu ajutorul senzorului ultrasonic.

```
#include "VarSpeedServo.h"
#include <NewPing.h>
#include <Servo.h>

VarSpeedServo RU;
VarSpeedServo RL;
VarSpeedServo LU;
VarSpeedServo LL;

NewPing sonar(4,3,200);

const int vel = 20, vel_Back = 10;
const int delay_Forward = 750, delay_Back = 1000;

const int array_cal[4] = {90,90,90,90};
int RU_Degree = 0, LU_Degree = array_cal[2] + 5;

const int num1 = 6;
const int array_forward[num1][4] =
{
  {0,-40,0,-20},
  {30,-40,30,-20},
  {30,0,30,0},
  {0,20,0,40},
  {-30,20,-30,40},
  {-30,0,-30,0},
};

const int num2 = 5;
const int array_turn[num2][4] =
{
  {-40,0,-20,0},
  {-40,30,-20,30},
  {0,30,0,30},
  {30,0,30,0},
  {0,0,0,0},
};

#define RUN

void Servo_Init()
{
  RU.attach(9);
  RL.attach(10);
  LU.attach(11);
  LL.attach(12);
}

void Adjust()
{
  for(RU_Degree = array_cal[0] - 5; RU_Degree <= array_cal[0]; RU_Degree += 1) {
    RU.write(RU_Degree);
    LU.write(LU_Degree--);
    delay(15);
  }
}

bool TooClose()
{
  int tooclose = 0;
  for(int a=0; a<5; a++) {
    delay(50);
    int din = sonar.ping_in();
    if (din < 7 && din > 0) tooclose++;
  }
  if (tooclose < 5) return 1;
  return 0;
}

void Forward()
{
  for(int x=0; x<num1; x++) {
    RU.slowmove (array_cal[0] + array_forward[x][0] , vel);
    RL.slowmove (array_cal[1] + array_forward[x][1] , vel);
    LU.slowmove (array_cal[2] + array_forward[x][2] , vel);
    LL.slowmove (array_cal[3] + array_forward[x][3] , vel);
    delay(delay_Forward);
  }
}

void Backward()
{
  for(int z=0; z<4; z++) {
    for(int y=0; y<num2; y++) {
      RU.slowmove (array_cal[0] + array_turn[y][0] , vel_Back);
      RL.slowmove (array_cal[1] + array_turn[y][1] , vel_Back);
      LU.slowmove (array_cal[2] + array_turn[y][2] , vel_Back);
      LL.slowmove (array_cal[3] + array_turn[y][3] , vel_Back);
      delay(delay_Back);
    }
  }
}

void setup()
{
#ifdef INSTALL
  Servo_Init();

  RU.slowmove (90 , vel);
  RL.slowmove (90 , vel);
  LU.slowmove (90 , vel);
  LL.slowmove (90 , vel);
  while(1);
#endif

#ifdef CALIBRATION
  Servo_Init();
  Adjust();

  RL.slowmove (array_cal[1] , vel);
  LL.slowmove (array_cal[3] , vel);
  delay(2000);
  while(1);
#endif

#ifdef RUN
  Servo_Init();
  Adjust();

  RL.slowmove (array_cal[1] , vel);
  LL.slowmove (array_cal[3] , vel);
  delay(2000);
#endif
}

void loop()
{
  while(TooClose()) Forward();
  Backward();
}
```


lenovo
IPS
4K

CHRISTMAS TREE

CODUL: Pentru dansul de lumini s-a realizat un cod cu ajutorul internetului de 2605 linii de cod, din acest motiv nu o să pot afișa codul în această prezentare!

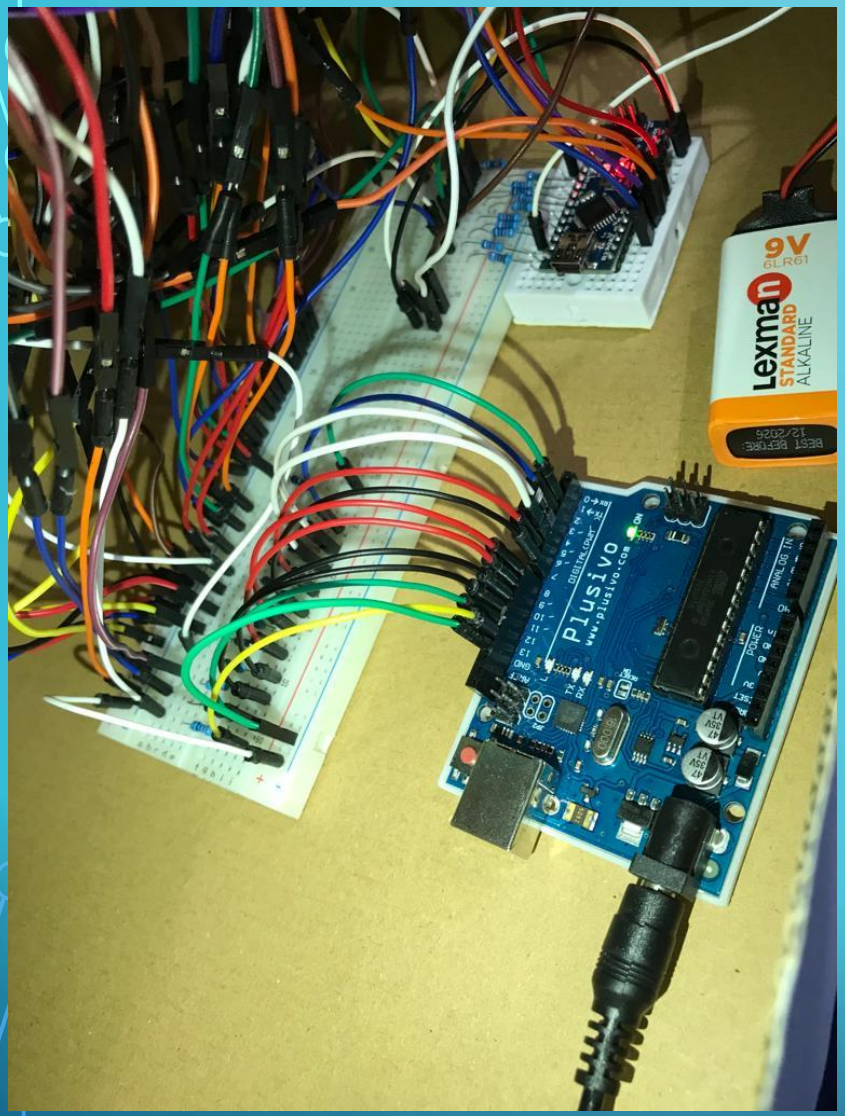
Mai jos se află o fotografie activă, dacă dați dublu click pe ea, se v-a deschide codul sursă (există posibilitatea să dea eroare la deschiderea lui pe un alt dispozitiv)!

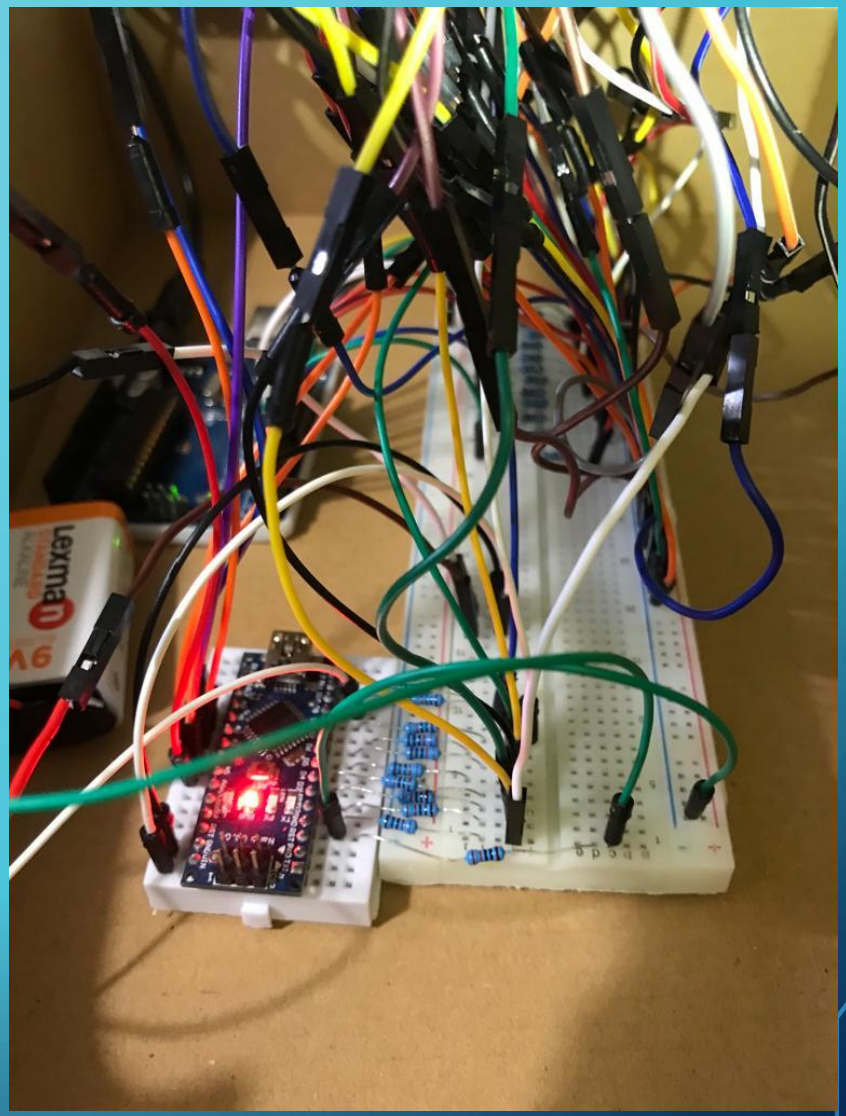
Christmas_Lights_LED.ino

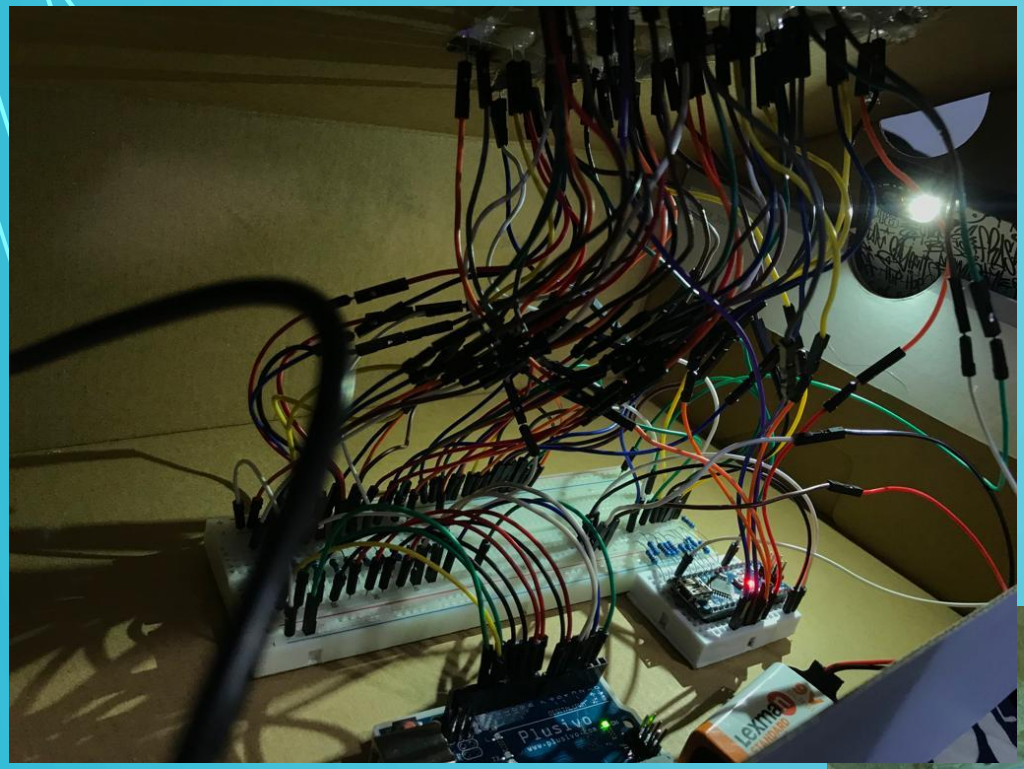
CODUL: Pentru cântecul Merry Christmas

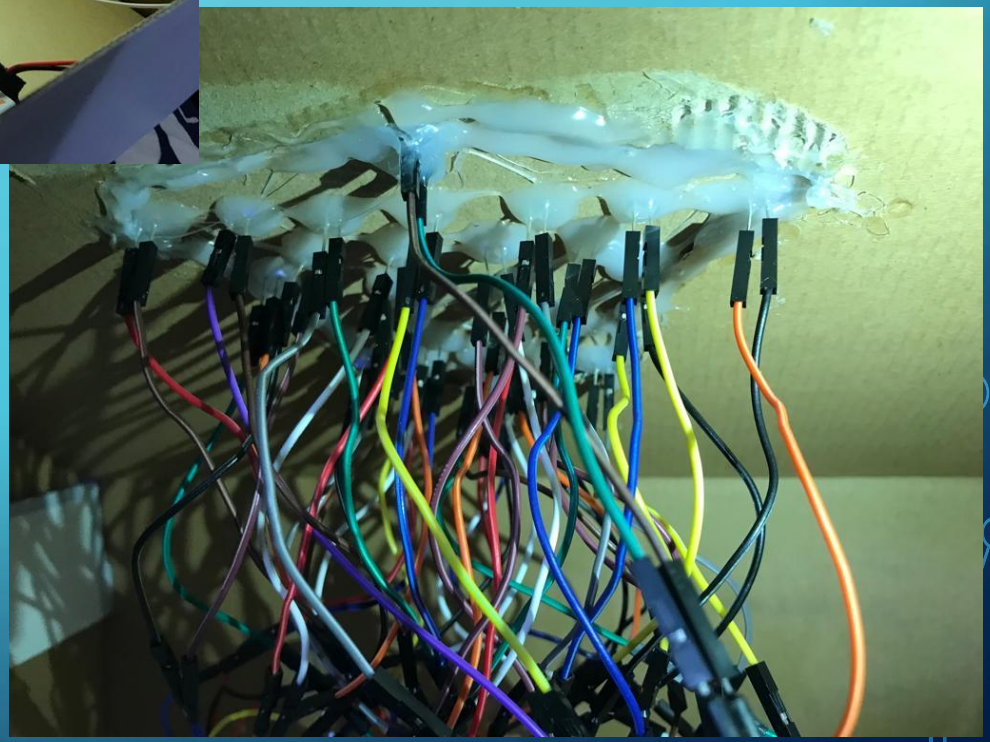
```
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78
#define NOTE_E2  82
#define NOTE_F2  87
#define NOTE_FS2 93
#define NOTE_G2  98
#define NOTE_GS2 104
#define NOTE_A2  110
#define NOTE_AS2 117
#define NOTE_B2  123
#define NOTE_C3  131
#define NOTE_CS3 139
#define NOTE_D3  147
#define NOTE_DS3 156
#define NOTE_E3  165
#define NOTE_F3  175
#define NOTE_FS3 185
#define NOTE_G3  196
#define NOTE_GS3 208
#define NOTE_A3  220
#define NOTE_AS3 233
#define NOTE_B3  247
#define NOTE_C4  262
#define NOTE_CS4 277
#define NOTE_D4  294
#define NOTE_DS4 311
#define NOTE_E4  330
#define NOTE_F4  349
#define NOTE_FS4 370
#define NOTE_G4  392
#define NOTE_GS4 415
#define NOTE_A4  440
#define NOTE_AS4 466
#define NOTE_B4  494
#define NOTE_C5  523
#define NOTE_CS5 554
#define NOTE_D5  587
#define NOTE_DS5 622
#define NOTE_E5  659
#define NOTE_F5  698
#define NOTE_FS5 740
#define NOTE_G5  784
#define NOTE_GS5 831
#define NOTE_A5  880
#define NOTE_AS5 932
#define NOTE_B5  988
#define NOTE_C6  1047
#define NOTE_CS6 1109
#define NOTE_D6  1175
#define NOTE_DS6 1245
#define NOTE_E6  1319
#define NOTE_F6  1397
#define NOTE_FS6 1480
#define NOTE_G6  1568
#define NOTE_GS6 1661
#define NOTE_A6  1760
#define NOTE_AS6 1865
#define NOTE_B6  1976
#define NOTE_C7  2093
#define NOTE_CS7 2217
#define NOTE_D7  2349
#define NOTE_DS7 2489
#define NOTE_E7  2637
#define NOTE_F7  2794
#define NOTE_FS7 2960
#define NOTE_G7  3136
#define NOTE_GS7 3322
#define NOTE_A7  3520
#define NOTE_AS7 3729
#define NOTE_B7  3951
#define NOTE_C8  4186
#define NOTE_CS8 4435
#define NOTE_D8  4699
#define NOTE_DS8 4978
#define REST     0
```

```
int tempo = 140;
int buzzer = 13;

int melody[] = {

 NOTE_C5,4, //1
 NOTE_F5,4, NOTE_F5,8, NOTE_G5,8, NOTE_F5,8, NOTE_E5,8,
 NOTE_D5,4, NOTE_D5,4, NOTE_D5,4,
 NOTE_G5,4, NOTE_G5,8, NOTE_A5,8, NOTE_G5,8, NOTE_F5,8,
 NOTE_E5,4, NOTE_C5,4, NOTE_C5,4,
 NOTE_A5,4, NOTE_A5,8, NOTE_ASS,8, NOTE_A5,8, NOTE_G5,8,
 NOTE_F5,4, NOTE_D5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,

 NOTE_F5,2, NOTE_C5,4, //8
 NOTE_F5,4, NOTE_F5,8, NOTE_G5,8, NOTE_F5,8, NOTE_E5,8,
 NOTE_D5,4, NOTE_D5,4, NOTE_D5,4,
 NOTE_G5,4, NOTE_G5,8, NOTE_A5,8, NOTE_G5,8, NOTE_F5,8,
 NOTE_E5,4, NOTE_C5,4, NOTE_C5,4,
 NOTE_A5,4, NOTE_A5,8, NOTE_ASS,8, NOTE_A5,8, NOTE_G5,8,
 NOTE_F5,4, NOTE_D5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, NOTE_C5,4,

 NOTE_F5,4, NOTE_F5,4, NOTE_F5,4,//17
 NOTE_E5,2, NOTE_E5,4,
 NOTE_F5,4, NOTE_E5,4, NOTE_D5,4,
 NOTE_C5,2, NOTE_A5,4,
 NOTE_ASS,4, NOTE_A5,4, NOTE_G5,4,
 NOTE_C6,8, NOTE_C5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, NOTE_C5,4,
 NOTE_F5,4, NOTE_F5,8, NOTE_G5,8, NOTE_F5,8, NOTE_E5,8,
 NOTE_D5,4, NOTE_D5,4, NOTE_D5,4,

 NOTE_G5,4, NOTE_G5,8, NOTE_A5,8, NOTE_G5,8, NOTE_F5,8, //27
 NOTE_E5,4, NOTE_C5,4, NOTE_C5,4,
 NOTE_A5,4, NOTE_A5,8, NOTE_ASS,8, NOTE_A5,8, NOTE_G5,8,
 NOTE_F5,4, NOTE_D5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, NOTE_C5,4,
 NOTE_F5,4, NOTE_F5,4, NOTE_F5,4,
 NOTE_E5,2, NOTE_E5,4,
 NOTE_F5,4, NOTE_E5,4, NOTE_D5,4,

 NOTE_C5,2, NOTE_A5,4,//36
 NOTE_ASS,4, NOTE_A5,4, NOTE_G5,4,
 NOTE_C6,4, NOTE_C5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, NOTE_C5,4,
 NOTE_F5,4, NOTE_F5,8, NOTE_G5,8, NOTE_F5,8, NOTE_E5,8,
 NOTE_D5,4, NOTE_D5,4, NOTE_D5,4,
 NOTE_G5,4, NOTE_G5,8, NOTE_A5,8, NOTE_G5,8, NOTE_F5,8,
 NOTE_E5,4, NOTE_C5,4, NOTE_C5,4,

 NOTE_A5,4, NOTE_A5,8, NOTE_ASS,8, NOTE_A5,8, NOTE_G5,8,//45
 NOTE_F5,4, NOTE_D5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, NOTE_C5,4,
 NOTE_F5,4, NOTE_F5,8, NOTE_G5,8, NOTE_F5,8, NOTE_E5,8,
 NOTE_D5,4, NOTE_D5,4, NOTE_D5,4,
 NOTE_G5,4, NOTE_G5,8, NOTE_A5,8, NOTE_G5,8, NOTE_F5,8,
 NOTE_E5,4, NOTE_C5,4, NOTE_C5,4,

 NOTE_A5,4, NOTE_A5,8, NOTE_ASS,8, NOTE_A5,8, NOTE_G5,8, //53
 NOTE_F5,4, NOTE_D5,4, NOTE_C5,8, NOTE_C5,8,
 NOTE_D5,4, NOTE_G5,4, NOTE_E5,4,
 NOTE_F5,2, REST,4
};

int notes = sizeof(melody) / sizeof(melody[0]) / 2;
int wholenote = (60000 * 4) / tempo;
int divider = 0, noteDuration = 0;
int thisNote = 0;

void setup() {
 pinMode(1, OUTPUT);
 digitalWrite(1, HIGH);
 pinMode(2, OUTPUT);
 digitalWrite(2, HIGH);
 pinMode(3, OUTPUT);
 digitalWrite(3, HIGH);
 pinMode(4, OUTPUT);
 digitalWrite(4, HIGH);
 pinMode(5, OUTPUT);
 digitalWrite(5, HIGH);
 pinMode(6, OUTPUT);
 digitalWrite(6, HIGH);
 pinMode(7, OUTPUT);
 digitalWrite(7, HIGH);
 pinMode(8, OUTPUT);
 digitalWrite(8, HIGH);
 pinMode(9, OUTPUT);
 digitalWrite(9, HIGH);
 pinMode(11, OUTPUT);
 digitalWrite(11, HIGH);
 pinMode(12, OUTPUT);
 digitalWrite(12, HIGH);
 pinMode(A0, OUTPUT);
 digitalWrite(A0, HIGH);
 pinMode(A1, OUTPUT);
 digitalWrite(A1, HIGH);
 pinMode(A2, OUTPUT);
 digitalWrite(A2, HIGH);
 pinMode(A3, OUTPUT);
 digitalWrite(A3, HIGH);
 pinMode(A4, OUTPUT);
 digitalWrite(A4, HIGH);
 pinMode(A5, OUTPUT);
 digitalWrite(A5, HIGH);

}

void loop() {
 // no need to repeat the melody.
 divider = melody[thisNote + 1];
  if (divider > 0) {
   noteDuration = (wholenote) / divider;
  } else if (divider < 0) {
   noteDuration = (wholenote) / abs(divider);
   noteDuration *= 1.5;
  }

  tone(buzzer, melody[thisNote], noteDuration * 0.9);

  delay(noteDuration);

  noTone(buzzer);
  thisNote = thisNote + 2;
}
```


9V
6LR61
Lexman
STANDARD
ALKALINE
BEST BEFORE:
12/2025
Plusivo
www.plusivo.com



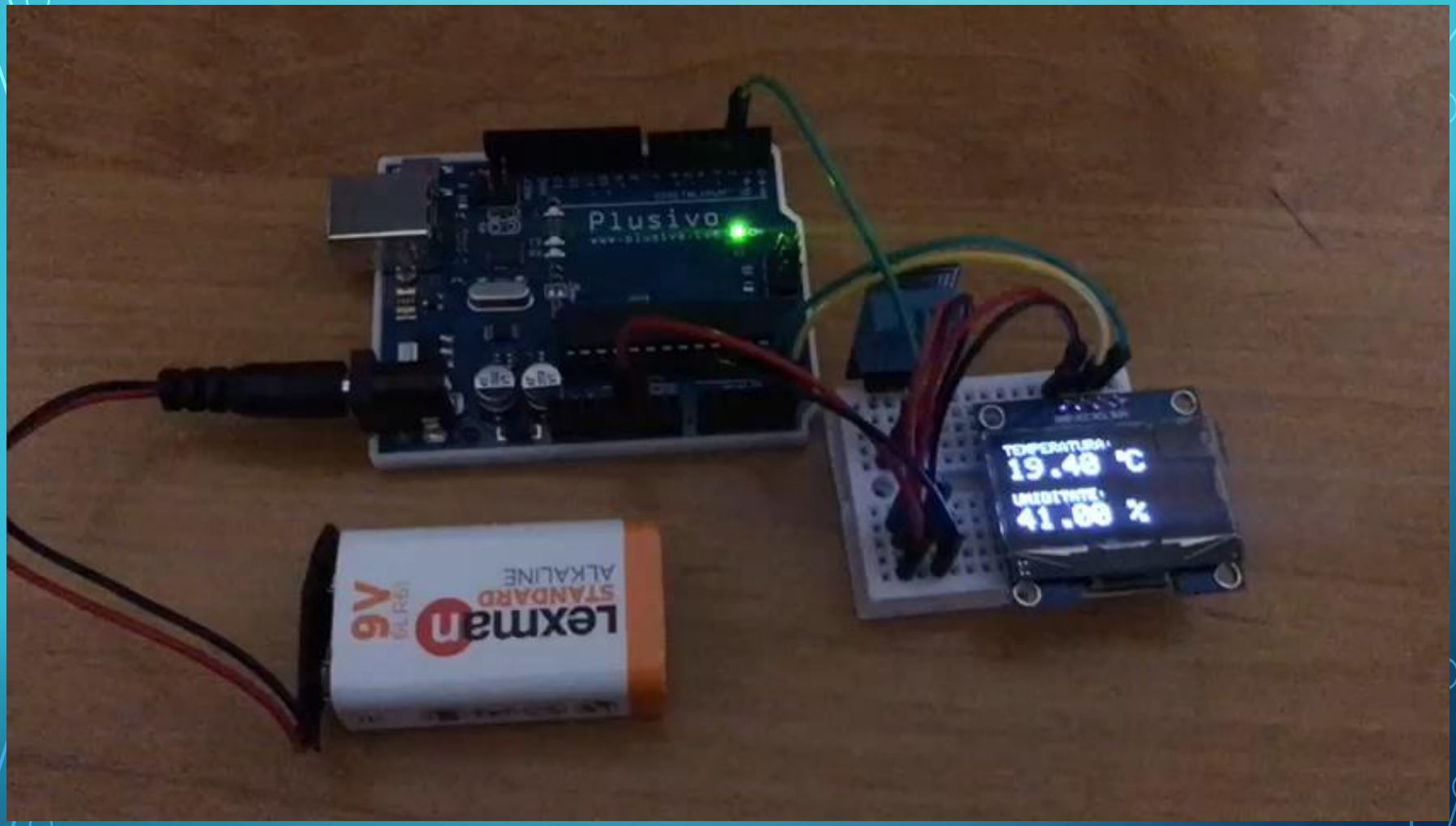


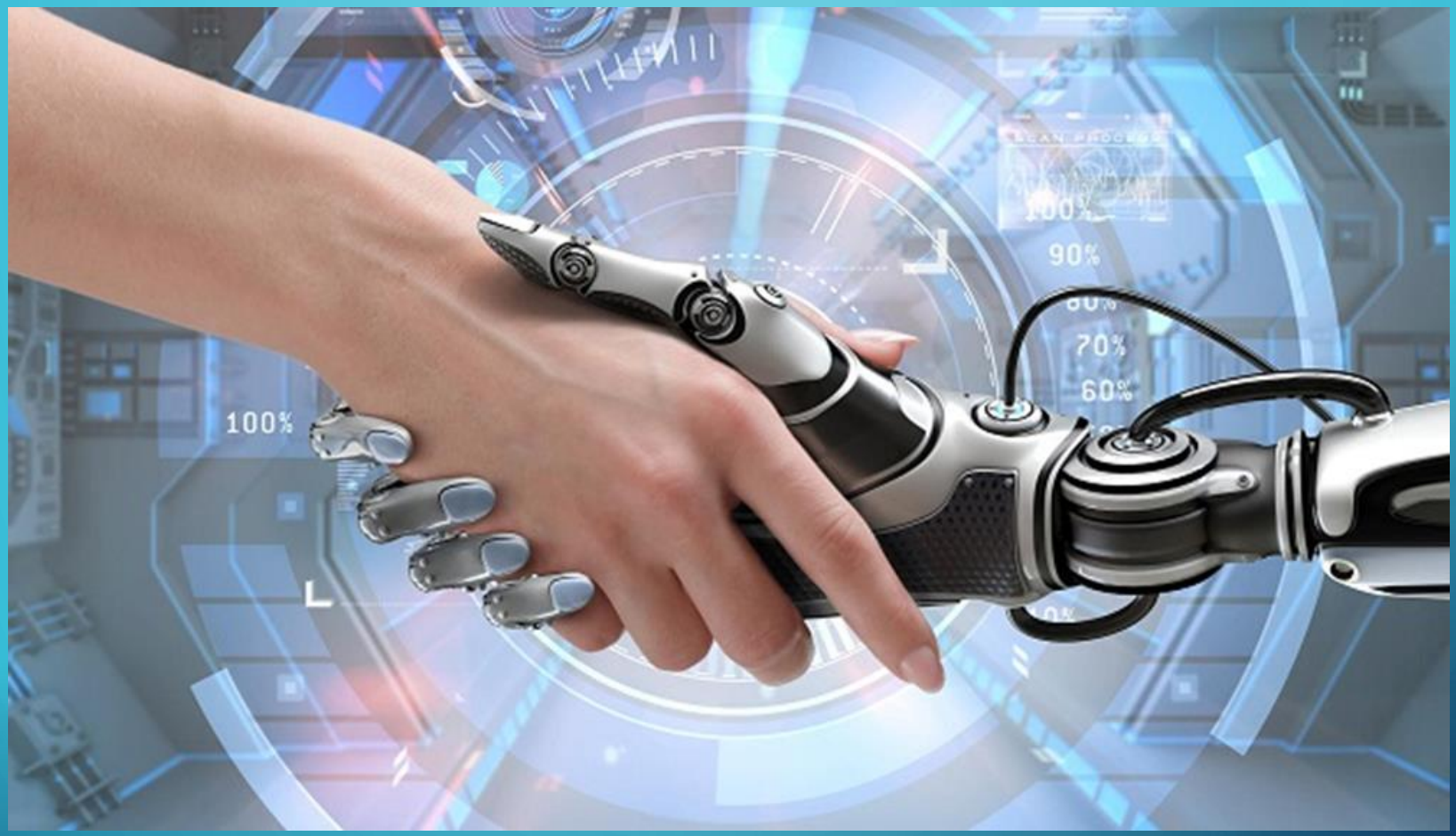


OLED PENTRU AFIȘAREA ÎN TIMP REAL A TEMPERATURII ȘI UMIDITĂȚII

CODUL:

```
#include "U8glib.h"
U8GLIB_SSD1306_128X64 u8g(U8G_I2C_OPT_NONE|U8G_I2C_OPT_DEV_0);          // I2C / TWI
#include <Arduino.h>
#include <dht.h>

dht DHT;
#define DHT22_PIN 8


void draw(void) {
  int chk = DHT.read22(DHT22_PIN);
  float temp=DHT.temperature;
  float humi=DHT.humidity;

  u8g.setFont(u8g_font_helvR24);
  u8g.setPrintPos(5,25);
  u8g.print(temp);
  u8g.setPrintPos(100,25);
  u8g.print("c");
  u8g.setPrintPos(5,60);
  u8g.print(humi);
  u8g.print("%");
  u8g.setFont(u8g_font_courB10);
  u8g.setPrintPos(90,8);
  u8g.print("o");
}

void setup(void) {

}

void loop(void) {

  // picture loop
  u8g.firstPage();
  do {
    draw();
  } while( u8g.nextPage() );

  // rebuild the picture after some delay
  delay(1000);
}
```


Plusivo
TEMPERATURA:
19.48 °C
41.00 %
lexman
STANDARD
ALKALINE
9V


100%
90%
70%
60%