

Traianis Eftenoiu: PET -> filament 3D

Prezentare proiect

Voi construi un dispozitiv care convertește PETurile de plastic uzate în filament pentru imprimante 3D, oferind o soluție ecologică și economică pentru reciclarea materialelor plastice și crearea de obiecte personalizate.

Descriere generală

Dispozitivul care face conversia este împărțit în 3 zone:

1. Zona de tăiere: face tăierea PETului într-o fâșie de plastic.
2. Zona de topire: face trecerea de la fâșia de plastic la filament.
3. Zona de strângere: filamentul scos din hotend va fi strâns pe o rolă care se rotește constant.



Hardware Design

Componente:

```
1x Arduino UNO
2x Breadboard
1x Stepper Nema17
1x Driver a4988
2x Potentiometru
1x Buton
1x LCD I2C
1x Hotend
1x IRFZ44N
1x S8050
2x Rezistoare 10k
1x Sursa 12V 2A (pt motor)
1x Sursa 12V 3A (pt hotend)
```



Software Design

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <thermistor.h>
#include <AccelStepper.h>
#include <SPI.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2); //16x2 LCD
thermistor therm1(A0,1); //Thermistor nr 1
AccelStepper stepper(1, 4, 5); //interface 1, pinDir 5, pinStep 4
```

```
//Pins
const int PWM_pin = 3;
const int motor_potentiometru_pin = A0;
const int temp_potentiometru_pin = A3;
const int adcPin = A2;
const int button = 2;
```

```
//Variables
float temp = 0.0;
float set_temperature = 200;
float PID_error = 0;
float previous_error = 0;
float elapsedTime, Time, timePrev;
int PID_value = 0;
bool set_temp = 0; //setting temp or not
bool to_clear = 0; //clear the lcd
```

```
int speed = 200;
```

```
//PID constants
int kp = 9.1;   int ki = 0.3;   int kd = 1.8;
int PID_p = 0;   int PID_i = 0;   int PID_d = 0;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  //Button init
  pinMode(button, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(button), temp_set, CHANGE);
```

```
  pinMode(PWM_pin,OUTPUT);
  TCCR2B = TCCR2B & B11111000 | 0x03;    // pin 3 and 11 PWM frequency of
  980.39 Hz
```

```
Time = millis();
```

```
//LCD init  
lcd.init();  
lcd.backlight();
```

```
//Stepper init  
stepper.setMaxSpeed(2000);
```

```
}
```

```
void loop() {
```

```
//Stepper zone  
int val_pot_motor = analogRead(motor_potentiometru_pin);  
Serial.println(val_pot_motor);  
speed = map(val_pot_motor,0,1023,0, 1000);  
stepper.setSpeed(speed);  
stepper.runSpeed();
```

```
//Hotend zone  
if(set_temp)  
{  
  if (to_clear)  
  {  
    lcd.clear();  
    to_clear = 0;  
  }  
  int val_pot_temp = analogRead(temp_potentiometru_pin);  
  set_temperature = map(val_pot_temp,0,1023,0,300);  
  lcd.setCursor(0,0);  
  lcd.print("Temp set to:");  
  lcd.setCursor(3,1);  
  lcd.print("T:");  
  lcd.setCursor(5,1);  
  lcd.print(set_temperature,1);  
}  
else  
{  
  if (to_clear)  
  {  
    lcd.clear();  
    to_clear = 0;  
  }  
}
```

```
temp = therm1.analog2temp(); // read temperature  
float total_temp = temp;  
//Approximate temp  
for (int i =0;i<9;i++)  
{  
  total_temp += therm1.analog2temp();
```

```

}
temp = total_temp/10;

```

```

//Next we calculate the error between the setpoint and the real value
PID_error = set_temperature - temp;
//Calculate the P value
PID_p = kp * PID_error;
//Calculate the I value in a range on +-3
if(-3 < PID_error <3)
{
    PID_i = PID_i + (ki * PID_error);
}
//For derivative we need real time to calculate speed change rate
timePrev = Time; // the previous time is stored
before the actual time read
Time = millis(); // actual time read
elapsedTime = (Time - timePrev) / 1000;
//Now we can calculate the D value
PID_d = kd*((PID_error - previous_error)/elapsedTime);
//Final total PID value is the sum of P + I + D
PID_value = PID_p + PID_i + PID_d;
//We define PWM range between 0 and 255
if(PID_value < 0)
{    PID_value = 0;    }
if(PID_value > 255)
{    PID_value = 255; }

```

```

//PWM signal to the mosfet on digital pin D3
analogWrite(PWM_pin,255-PID_value);
previous_error = PID_error;

```

```

lcd.setCursor(0,0);
lcd.print("PID TEMP control");
lcd.setCursor(0,1);
lcd.print("S:");
lcd.setCursor(2,1);
lcd.print(set_temperature,1);
lcd.setCursor(9,1);
lcd.print("R:");
lcd.setCursor(11,1);
lcd.print(temp,1);
}
delay(1000);

```

```

}

void temp_set() {

```

```

set_temp = !set_temp;
to_clear = 1;

```


}

Rezultate Obținute

Care au fost rezultatele obținute în urma realizării proiectului vostru.

Concluzii

Download

O arhivă (sau mai multe dacă este cazul) cu fișierele obținute în urma realizării proiectului: surse, scheme, etc. Un fișier README, un ChangeLog, un script de compilare și copiere automată pe uC crează întotdeauna o impresie bună .

Fișierele se încarcă pe wiki folosind facilitatea **Add Images or other files**. Namespace-ul în care se încarcă fișierele este de tipul **:pm:prj20??:c?** sau **:pm:prj20??:c?:nume_student** (dacă este cazul).
Exemplu: Dumitru Alin, 331CC → **:pm:prj2022:cc:dumitru_alin**.

Jurnal

19.05.2024: Am reusit sa fac filament din fasia de plastic, dar trebuie sa schimb motorul dc cu un motor pas cu pas, actualul neavand destula putere pentru a rotii rotile la o viteza mica.



Bibliografie/Resurse

Listă cu documente, datasheet-uri, resurse Internet folosite, eventual grupate pe **Resurse Software** și **Resurse Hardware**.

[Export to PDF](#)

From:

<http://ocw.cs.pub.ro/courses/> - **CS Open CourseWare**

Permanent link:

<http://ocw.cs.pub.ro/courses/pm/prj2024/ddosaru/traianis.eftenoiu>



Last update: **2024/05/25 21:10**