Monday, February 22, 2021 12:34 AM

Graphite control project.

Antecedents.

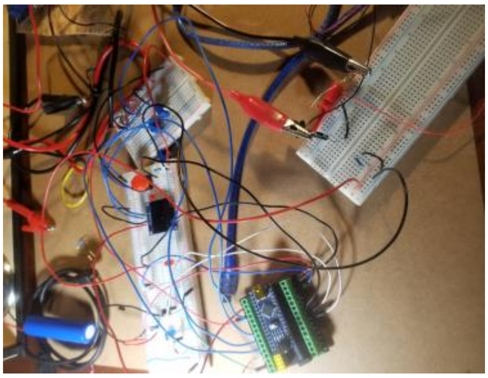
Originally each time that the chamber got empty--it is detected by a sensor inserted in the chamber- the valve on the hopper opened by a fixed time of five-second and then it is closed. Sometimes the quantity of graphite deposited on the chamber is not enough and the run table stops with the foam robot over the tools without completes the pouring process. It is traduced in a waste of time and generation of scrap and contamination of the tools. One of the objectives of this project is to provide a method that allows changing this time to avoid these issues.

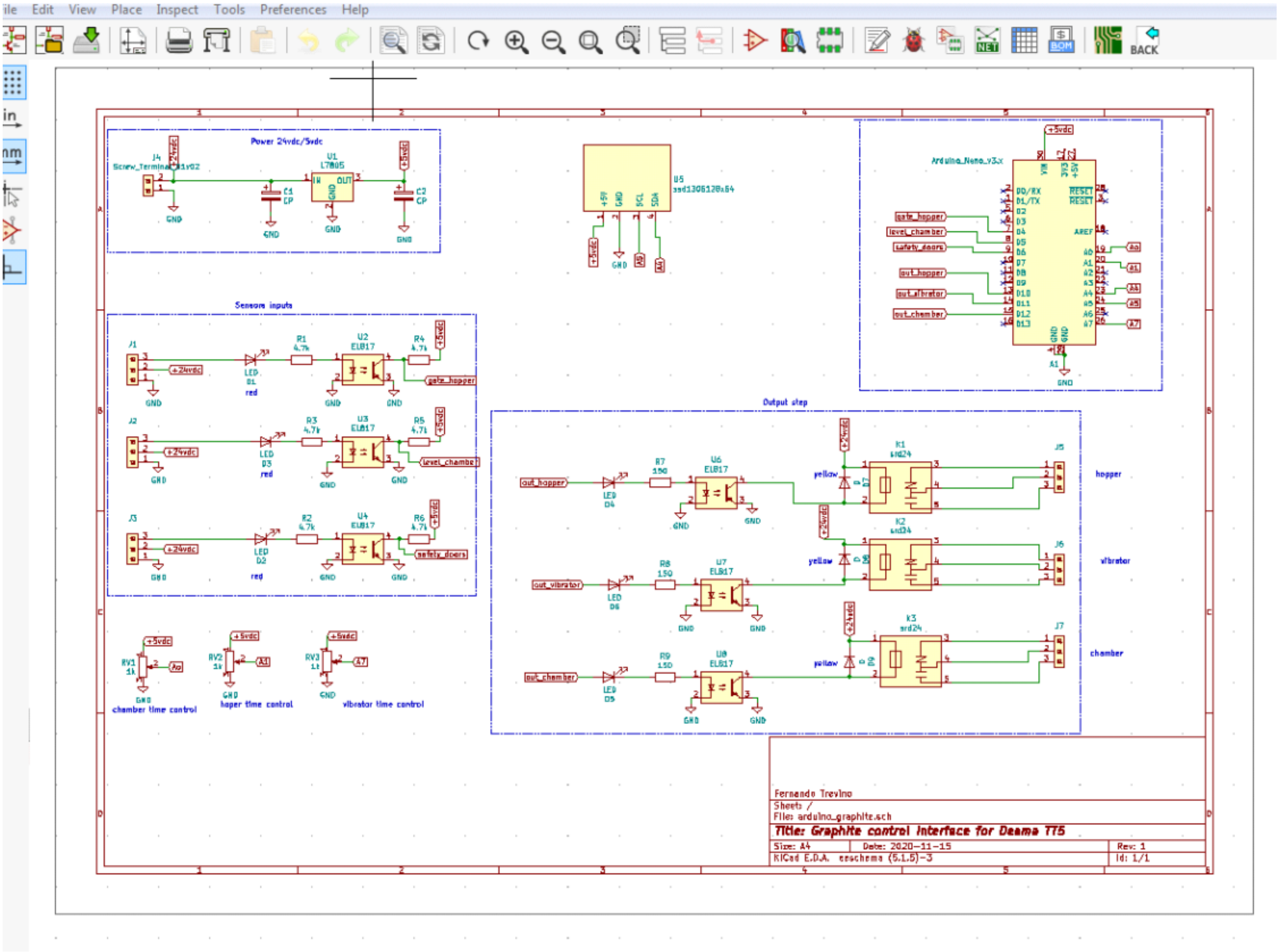
Additionally, it was suggested by Alex to provide the graphite control with the ability to regulate the time for refill the hopper.
The reason is that originally each time the hopper got empty, The frequency drive is activated and it is kept on only for a short time while the gate is closed. It is traduced mainly on more start times for the motor(less live of service) and more consumption of energy(a motor uses
from 5 to 12 times its nominal current each time it is started) .Also, the hopper is filled at less than its 40% of capacity in these conditions.

Implementation

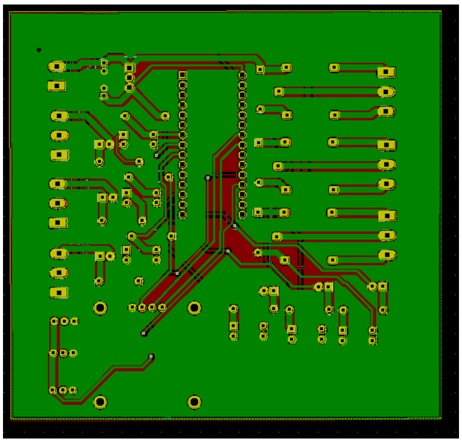
It was selected to use an Arduino nano as a main control because it has all the necessary potential in itself for handle monitorization and time sequences. Also, it is possible to expand the system for future requirements in a relatively easy way.

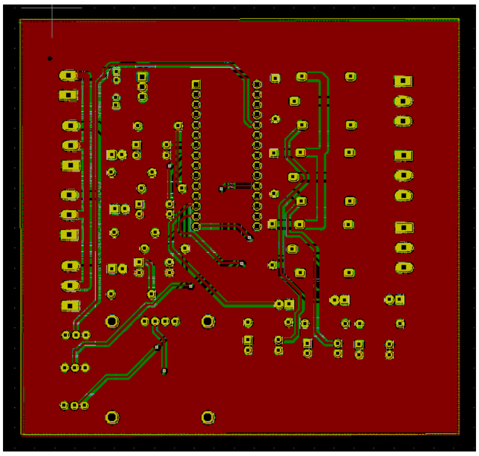
It was used the free software KICAD for the creation of the electronic modules and verified its functionality on protoboards

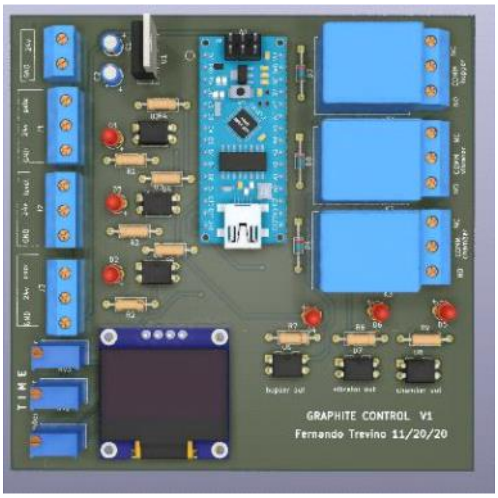




After this ,it was generated the printed boards and its simulation.







Code for programming Arduino Nano:
I wrote the next code using the concept of the state machines, incorporating some libraries , subroutines and functions.

```
/
*********************************************************************************
**************************************************************
*             Usando mi inspiracion
*             10/19/2020 Michigan domingo por la noche
*             10/22/2020 agregando screen para ver las entradas analogicas de los
 pots.
*             11/11/20 Agregando funcion de desactivacion cuando puertas a jaula
de robot son abiertas.
*********************************************************************************
****************************************************************/
//            Librerias para screen

#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
// Declaration for an SSD1306 display connected to I2C (SDA, SCL pins)
#define OLED_RESET     -1 //4 Reset pin # (or -1 if sharing Arduino reset pin)
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);


//Variables
boolean sensor_hopper;
boolean sensor_chamber;
boolean someDoorOpen=0;// Detector de Puertas en jaula robot
long lectura_pot_ch=0; //Para valores analogos chamber A0
long lectura_pot_h=0; //Para valores analogos hopper  A1
long lectura_pot_v=0; //Para valores analogos vibrador A7


//Configuracion de pines del arduino nano
 const int gate_hopper=4;
 const int level_chamber=5;
 const int safety_doors=6;
 const int output_hopper=10;
 const int output_chamber=12;
 const int output_vibrador=11;
//****** Temporizado de las tareas *********
// chamber
unsigned long tiempo_inicial=0;
unsigned long tiempo_transcurrido=0;
int chamber_bussy=0;
unsigned long timer_chamber=0;//Se actualizara con el valor del pot
 // hopper
unsigned long tiempo_inicial_h=0;
unsigned long tiempo_transcurrido_h=0;
int hopper_bussy=0;
unsigned long timer_hopper=30000; //Se actualizara con el valor del pot

// vibrador
unsigned long tiempo_inicial_v=0;
unsigned long tiempo_transcurrido_v=0;
int vibrator_bussy=0;
long timerv_on=1000; //Tiempo en estado on,modificable atraves de analogica 7
const long timerv_off=2000;//tiempo en estado off

//==============Declarando prototipo de funciones ======================
void ReadPot();
void Hopper_on();
void Hopper_off();
void Chamber_on();
void Chamber_off();
void Vibrator_on();
void Vibrator_off();
void pantalla();
void manejador_de_tareas();
//
================================================================================
=================================================
void setup(){
//Serial.begin(9600);
/*================================================================*/
pinMode(safety_doors,INPUT);//puertas de entrada a jaula robot
pinMode(gate_hopper,INPUT);
pinMode(output_hopper,OUTPUT);
digitalWrite(output_hopper,LOW);
pinMode(level_chamber,INPUT);
pinMode(output_chamber,OUTPUT);
digitalWrite(output_chamber,LOW);
pinMode(output_vibrador,OUTPUT);
digitalWrite(output_vibrador,LOW);
//
=====================================S C R E E N  128 X 64  =====================
===================================
 // SSD1306_SWITCHCAPVCC = generate display voltage from 3.3V internally
display.begin(SSD1306
_SWITCHCAPVCC, 0x3C);//I changed to 0x3C... Address 0x3D for 128x64
// Clear the buffer
display.clearDisplay();
//Usando Texto
display.setTextSize(1.8);
display.setTextColor(WHITE);
display.setCursor(0,0);
display.clearDisplay();
display.setCursor(10,0);//(columna,renglon)
display.println("Timers     seconds");
display.display();
display.setTextSize(1.6);
display.setCursor(10,20);
display.println("Chamber:");
display.setCursor(10,35);
display.println("Hopper:");
display.setCursor(10,50);
display.println("Vibrator:");
}
//
================================================================================
===================================================
void loop(){
 manejador_de_tareas();
}
//
================================================================================
===================================================
/
* ========                                              F U N S I O N E S Y P R O
 C E S O S                   ===================== */




void Hopper_on(){
/* Is being used an opto isolator EL718 so when the sensor is high=input on opto
, the output at transistor is lower= sensor_hopper*/
sensor_hopper=digitalRead(gate_hopper);
    if (!sensor_hopper){
            digitalWrite(output_hopper,HIGH);
            hopper_bussy=1;//Establecer indicador de ocupado
            tiempo_inicial_h=millis();//Anotar tiempo de inicio
            Vibrador_on();


   }




}

void Hopper_off(){

  digitalWrite(output_hopper,LOW);
   hopper_bussy=0;
}
//
================================================================================
===================================================
void Chamber_on(){
      someDoorOpen=digitalRead(safety_doors);
      if (someDoorOpen) {// Si las puertas estan cerradas =1
        sensor_chamber=digitalRead(level_chamber);

          /
* The sensor on mix head is off when it detect lower level on graphite */
        /
* it is being used opto isolator E817 so when sensor is off ,the output at transi
stor is on */
                     if (sensor_chamber){
                             digitalWrite(output_chamber,HIGH);
                             chamber_bussy=1;
                             tiempo_inicial=millis();
                    }
      }
```

```
}

void Chamber_off(){
  //Despues del temporizado
  digitalWrite(output_chamber,LOW);
  chamber_bussy=0;

  }
//
=================================================================================
====================================================
  void Vibrador_on(){
     digitalWrite(output_vibrador,HIGH);
    vibrator_bussy=1;
    tiempo_inicial_v=millis();

    }

  void Vibrador_off(){
   digitalWrite(output_vibrador,LOW);
  vibrator_bussy=0;
  tiempo_inicial_v=millis();
    }
/
*********************************************************************************
**********************************/
void manejador_de_tareas(){
  //Iniciar tareas
  ReadPot();
  Chamber_on();
  Hopper_on();
  pantalla();
  //================ CONTROL DE TIEMPOS  =========*  =========
  /*   PARA CHAMBER */
  switch (chamber_bussy){
    case 0:
    break;
    case 1://La camara se esta llenando

    tiempo_transcurrido= millis()-tiempo_inicial;
    if(tiempo_transcurrido>=timer_chamber){
        Chamber_off();
      }
    break;
    }
 //===========================================================
 /*   PARA HOPPER */
  switch (hopper_bussy){
    case 0:
    //No esta en funcion
    break;
    case 1://La Tolva se esta llenando
    tiempo_transcurrido_h= millis()-tiempo_inicial_h;
    if(tiempo_transcurrido_h>=timer_hopper){
        Hopper_off();
      }else{
        //Checar si se apago el vibrador para encenderlo nuevamente
         tiempo_transcurrido_v=millis()-tiempo_inicial_v;
        if(vibrator_bussy==0 && tiempo_transcurrido_v>=timerv_off)
{Vibrador_on();}
        }
    break;
    }
  //===========================================================
  /*   PARA VIBRADOR */
  switch (vibrator_bussy){
    case 0:
    break;
    case 1://vibrador funcionando
    tiempo_transcurrido_v= millis()-tiempo_inicial_v;
    if(tiempo_transcurrido_v>=timerv_on){
        Vibrador_off();
      }
    break;
    }

  }


//
=================================================================================
====================================================

void pantalla(){



 //display.clearDisplay(); //borra los titulos
  display.setTextSize(1.6);
  display.setCursor(80,20);
  //Para evitar el amontonamiento de pixeles
  display.setTextColor(WHITE,BLACK);//puede resivir dos parametros:color de letra
,fondo
  display.println( String(lectura_pot_ch));

  display.setCursor(80,35);
  display.setTextColor(WHITE,BLACK);//
  display.println(String(lectura_pot_h));
  display.setCursor(80,50);
  display.setTextColor(WHITE,BLACK);//
  display.println(String(lectura_pot_v));
  display.display();
    }
 //
================================================================================
=============
void ReadPot(){
  lectura_pot_ch=map(analogRead(A0),0,1023,0,20);//Reed  analogic input A0 ,time
on miliseconds,
  timer_chamber=1000
*lectura_pot_ch;  //now time is in seconds,maximum is 20 seconds

  lectura_pot_h=map(analogRead(A1),0,1023,10,600);//Leer entrada analogica A1 hop
per
  timer_hopper=1000*lectura_pot_h;//maximum 6 minutes.
  //Vibrador
 lectura_pot_v=map(analogRead(A7),0,1023,0,20);//Leer entrada analogica A7
 timerv_on=1000* lectura_pot_v;
  }
```

Finally it was integrated all the components on the pc board and tested it.