```
/**
   @brief Programa para el control de la maqueta de convertidores
DC/DC.

   Este programa es para el microcontrolador encargado del
control de la maqueta de convertidores DC/DC.

   @file control.ino
   @authors Hector Bohe 1491407@campus.euss.org
   @date 14.06.2022

   @copyright GNU Public License.
   @section LICENSE
   This program is free software; you can redistribute it and/or
modify
   it under the terms of the GNU General Public License as
published by
   the Free Software Foundation; either version 2 of the License,
or
   (at your option) any later version.

   @section  Referencias
    Referencias
      - Display ->
https://naylampmechatronics.com/blog/35_tutorial-lcd-con-i2c-contr
ola-un-lcd-con-solo-dos-pines.html
      - Señales PWM -> https://www.laboratoriogluon.
com/generar-senal-pwm-para-servo-con-avr-atmega328p/
      - General -> https://www.arduino.cc/reference/es/
      - Datasheet ->
http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48A-PA-88A-P
A-168A-PA-328-P-DS-DS40002061A.pdf

 **/

#include <Wire.h> //Biblioteca comunicación I2C/TWI
#include <LiquidCrystal_I2C.h> //Biblioteca comunicación diplay
liquido con I2C.

//Crear el objeto lcd  dirección  0x27 y 16 columnas x 2 filas
```

```
LiquidCrystal_I2C lcd(0x27, 16, 2); //

// Definicoñon de pines
int inModo1 = 14; // Pin 1 configuracion de modo
int inModo2 = 3; // Pin 2 configuracion de modo
int inModo3 = 11; // Pin 3 configuracion de modo
int aD = 6; // Pin analógico D
int afr = 1; // Pin analógico frecuencia
int ailbuck = 2; // Pin analógico IL_BUCK
int avobuck = 3; // Pin analógico Vo_BUCK
int ailboost = 4; // Pin analógico IL_BOOST
int avoboost = 5; // Pin analógico Vo_BOOST
int PWM_BUCK = 10; // Pin PWM_BUCK
int PWM_BOOST = 9; // Pin PWM_BOOST
int RL_1 = 2; //Pin resitencia de carga boost 10ohm
int RL_2 = 4; //Pin resitencia de carga boost 20ohm
int RL_3 = 7; //Pin resitencia de carga boost 39ohm
int RL_4 = 8; //Pin resitencia de carga buck 10ohm
int RL_5 = 12; //Pin resitencia de carga buck 5ohm
int RL_6 = 13; //Pin resitencia de carga buck 5ohm

//Variables
String convertidor = ""; // Mensaje convertidor [BUCK, BOOST]
String modo = ""; // Mensaje modo [CCM, DCM, LIM]

int modo0 = 0; // Todos los modos
int modo1 = 0; // Modo 1
int modo2 = 0; // Modo 2
int modo3 = 0; // Modo 3

int D = 0; // Ciclo de trabajo
int D_100 = 0; // Ciclo de trabajo en %
int f;
String fr = "";
String RL = "";
int top = 511;
int top_old = 511;
int comparacion = 256;

int ilbuck = 2; // Lectura IL_BUCK
```

```
int vobuck = 3; // Lectura Vo_BUCK
int ilboost = 4; // Lectura IL_BOOST
int voboost = 5; // Lectura Vo_BOOST

//Caracter LCD //Omega mayuscula
byte omega[8] = {
  B00000,
  B01110,
  B10001,
  B10001,
  B10001,
  B01010,
  B11011,
  B00000
};

void setup() {
  // Configuración LCD
  lcd.init();
  lcd.backlight(); // Se enciende la luz de fondo.
  lcd.createChar (0, omega); // Se crea los nuevos caracteres
  lcd.clear(); // Se limpia el display

  // Se configuran los pines PWM
  DDRB |= ( 1 << PB1 ); // Se configura el PB1 (D9) como salida.
  DDRB |= ( 1 << PB2 ); // Se configura el PB2 (D10) como salida.
  TCNT1 = 0; // Se reinicia el contador
  ICR1 = 511; // CSe configura el periodo de la señal (el TOP de
nuestra PWM)
  TCCR1A =  (1 << COM1A1) | (0 << COM1A0) ; // Se configura a
'LOW' el OCR1A cuando coincida el Compare Match
  TCCR1A |=  (1 << WGM11) | (0 << WGM10) ; // Fast PWM: TOP: ICR1
  TCCR1B = (1 << WGM13) | (1 << WGM12); // // Fast PWM: TOP: ICR1
  TCCR1B |= (0 << CS12) | (0 << CS11) | ( 1 << CS10 ); // Se
configura el preescaler a 0

  //Se configuran los pines
  pinMode(inModo1, INPUT);    // Definimos el pin 14 como entrada
  pinMode(inModo2, INPUT);    // Definimos el pin 3 como entrada
  pinMode(inModo3, INPUT);    // Definimos el pin 11 como entrada
```

```
  pinMode(RL_1, OUTPUT);    // Definimos el pin 2 como salida
  pinMode(RL_2, OUTPUT);    // Definimos el pin 4 como salida
  pinMode(RL_3, OUTPUT);    // Definimos el pin 7 como salida
  pinMode(RL_4, OUTPUT);    // Definimos el pin 8 como salida
  pinMode(RL_5, OUTPUT);    // Definimos el pin 12 como salida
  pinMode(RL_6, OUTPUT);    // Definimos el pin 13 como salida
}

void loop() {

  f = analogRead(afr); // Lectura potenciometro frecuencia
  f = map(f, 0, 1000, 1, 10); // Se cambia de rango f

  // Segun posición del potenciometro se selecciona una frecuencia
  switch (f) {
    case 1:
      top = 399;
      fr = "40kHz";
      break;
    case 2:
      top = 499;
      fr = "32kHz";
      break;
    case 3:
      top = 511;
      fr = "31.25kHz";
      break;
    case 4:
      top = 639;
      fr = "25kHz";
      break;
    case 5:
      top = 799;
      fr = "20kHz";
      break;
    case 6:
      top = 999;
      fr = "16kHz";
      break;
    case 7:
```

```
      top = 1279;
      fr = "12.5kHz";
      break;
    case 8:
      top = 1599;
      fr = "10kHz";
      break;
    case 9:
      top = 1999;
      fr = "8kHz";
      break;
    case 10:
      top = 2559;
      fr = "6.25kHz";
      break;
  }

  // Si se detecta un cambió respecto la frecuencia anterior se
modifica esta
  if (top != top_old) {
    top_old = top;
    ICR1 = top; // Configuramos el TOP del PWM
  }

  D = analogRead(aD); // Se lee el potenciometro del ciclo de
trabajo
  D = map(D, 0, 1023, 0, top + 1); // Se cambia de rango D
  D_100 = map(D, 0, top + 1, 0, 100); // Se cambia de rango D en %

  modo1 = digitalRead(inModo1); // Lectura del pin modo1
  modo2 = digitalRead(inModo2); // Lectura del pin modo2
  modo3 = digitalRead(inModo3); // Lectura del pin modo3
  modo0 = (modo1 * 4) + (modo2 * 2) + (modo3 * 1); // Se calcula
el modo de funcionamiento

  // Selección modo de funcionamiento
  switch (modo0) {
    case 2: // Buck modo CCM
      convertidor = "Buck ";
      modo = "CCM";
```

```
    RL = " 2";
    analogWrite(PWM_BUCK, D);
    analogWrite(PWM_BOOST, 0);
    digitalWrite(RL_1, LOW);
    digitalWrite(RL_2, LOW);
    digitalWrite(RL_3, LOW);
    digitalWrite(RL_4, HIGH);
    digitalWrite(RL_5, HIGH);
    digitalWrite(RL_6, HIGH);
    break;
  case 1: // Buck modo DCM
    convertidor = "Buck ";
    modo = "DCM";
    RL = "10";
    analogWrite(PWM_BUCK, D);
    analogWrite(PWM_BOOST, 0);
    digitalWrite(RL_1, LOW);
    digitalWrite(RL_2, LOW);
    digitalWrite(RL_3, LOW);
    digitalWrite(RL_4, HIGH);
    digitalWrite(RL_5, LOW);
    digitalWrite(RL_6, LOW);
    break;
  case 3: // Buck caso limite
    convertidor = "Buck ";
    modo = "LIM";
    RL = " 5";
    analogWrite(PWM_BUCK, D);
    analogWrite(PWM_BOOST, 0);
    digitalWrite(RL_1, LOW);
    digitalWrite(RL_2, LOW);
    digitalWrite(RL_3, LOW);
    digitalWrite(RL_4, LOW);
    digitalWrite(RL_5, HIGH);
    digitalWrite(RL_6, LOW);
    break;
  case 6: // Boost modo CCM
    convertidor = "Boost";
    modo = "CCM";
    RL = " 8";
```

```
    analogWrite(PWM_BOOST, D);
    analogWrite(PWM_BUCK, 0);
    digitalWrite(RL_1, HIGH);
    digitalWrite(RL_2, LOW);
    digitalWrite(RL_3, HIGH);
    digitalWrite(RL_4, LOW);
    digitalWrite(RL_5, LOW);
    digitalWrite(RL_6, LOW);
    break;
  case 5: // Boost modo DCM
    convertidor = "Boost";
    modo = "DCM";
    RL = "20";
    analogWrite(PWM_BOOST, D);
    analogWrite(PWM_BUCK, 0);
    digitalWrite(RL_1, LOW);
    digitalWrite(RL_2, HIGH);
    digitalWrite(RL_3, LOW);
    digitalWrite(RL_4, LOW);
    digitalWrite(RL_5, LOW);
    digitalWrite(RL_6, LOW);
    break;
  case 7: // Boost caso limite
    convertidor = "Boost";
    modo = "LIM";
    RL = "10";
    analogWrite(PWM_BOOST, D);
    analogWrite(PWM_BUCK, 0);
    digitalWrite(RL_1, HIGH);
    digitalWrite(RL_2, LOW);
    digitalWrite(RL_3, LOW);
    digitalWrite(RL_4, LOW);
    digitalWrite(RL_5, LOW);
    digitalWrite(RL_6, LOW);
    break;
  default: // OFF
    convertidor = "OFF";
    modo = "";
    RL = "        ";
    analogWrite(PWM_BUCK, 0);
```

```
      analogWrite(PWM_BOOST, 0);
      digitalWrite(RL_1, LOW);
      digitalWrite(RL_2, LOW);
      digitalWrite(RL_3, LOW);
      digitalWrite(RL_4, LOW);
      digitalWrite(RL_5, LOW);
      digitalWrite(RL_6, LOW);
      break;
  }

  ilbuck = analogRead(ailbuck); // Lectura corriente bobina buck
  vobuck = analogRead(avobuck); // Lectura tensión salida buck
  ilboost = analogRead(ailboost); // Lectura corriente bobina
boost
  voboost = analogRead(avoboost); // Lectura tensión salida boost

  // Mostramos datos por el LCD
  lcd.setCursor(0, 0);// Ubicamos el cursor en la primera
posición(columna:0) de la primera línea
  lcd.print(convertidor);
  lcd.print("  ");
  lcd.print(modo);
  lcd.print("   ");
  lcd.print(RL);
  lcd.write (byte (0));
  lcd.print("               ");

  lcd.setCursor(0, 1); // Ubicamos el cursor en la primera
posición(columna:0) de la segunda línea
  lcd.print(fr);
  lcd.print("     ");
  lcd.print(D_100);
  lcd.print("%");
  lcd.print("               ");
}
```