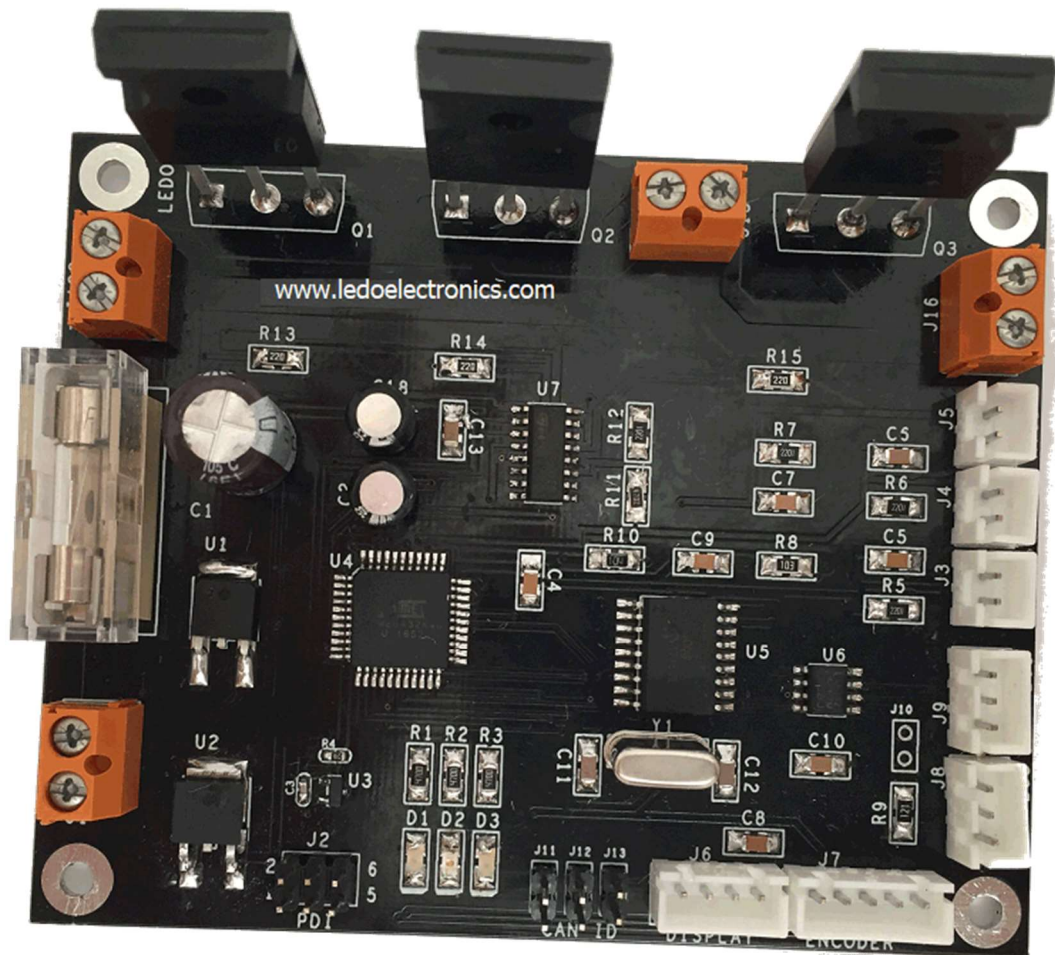


Controlador de temperatura PID con CAN bus



- **Voltaje de alimentación: 12...24V**
- **Tres canales PID de 12V/24V 6A DC**
- **NTC como sensores de temperatura**
- **Conector para Rotary Encoder**
- **Conector para Display I2C**
- **Interface CAN Bus (Can Open compatible)**
- **Xmega32a4u programmable in system.**

El modulo cuenta con el controlador Can MCP2515 compatible con el protocolo CAN V2.0B.

El Controlador Xmega32a4u se encarga de generar los pulsos PWM para el control de los elementos calefactores / enfriadores, acorde con algoritmo de control PID, y

además contiene el protocolo de comunicación. Los Jumper J7, J8, J9 permiten fijar una dirección en el bus para compatibilidad con el protocolo **Can Open**. Los LEDs D1, D2 y D3 sirven de indicación de estado.

El control puede realizarse de forma remota mediante CAN bus, o localmente haciendo uso de un encoder y un display I2C.

En el conector J6 puede ser conectada una pantalla Oled I2C con fines de diagnóstico y visualización de variables.

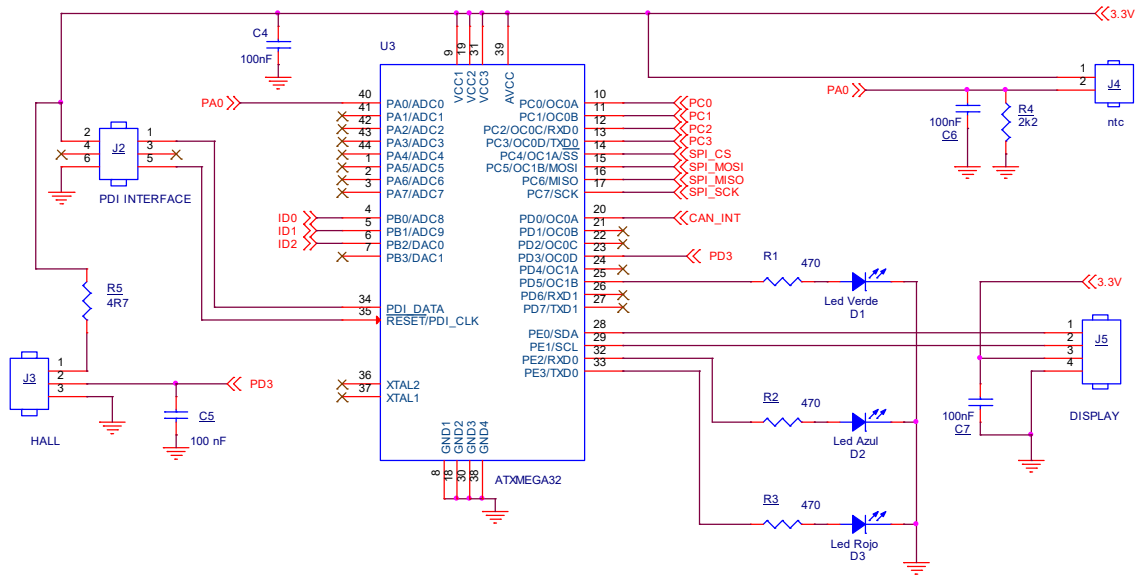


Fig.1. CPU.

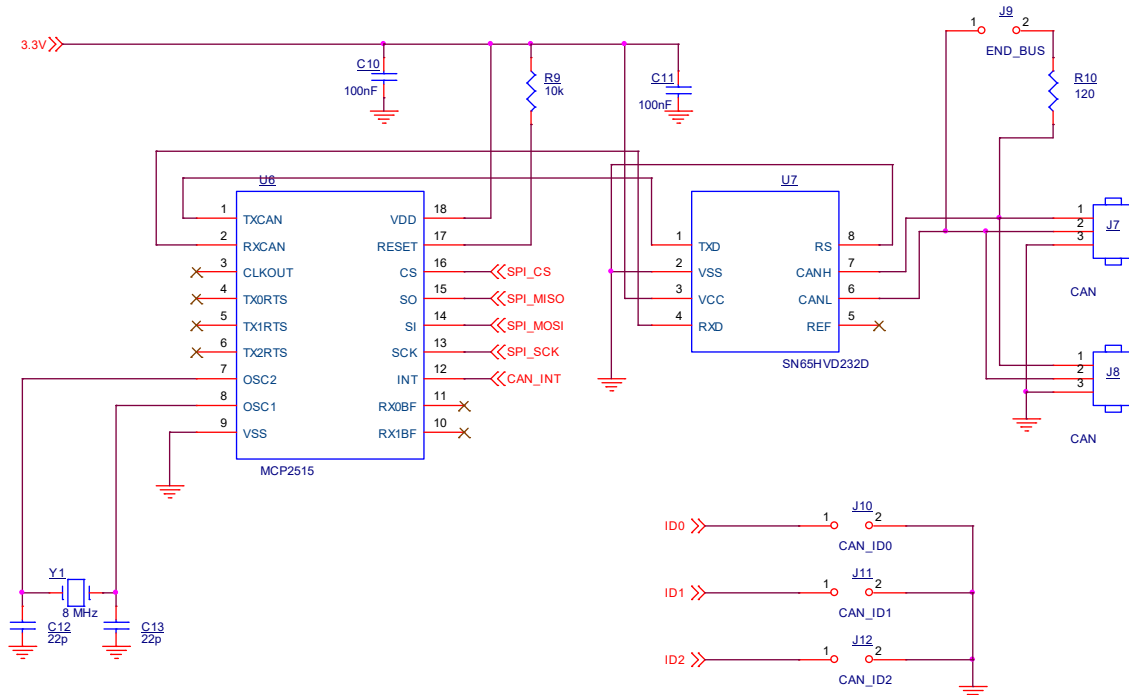


Fig.2. CAN Bus.

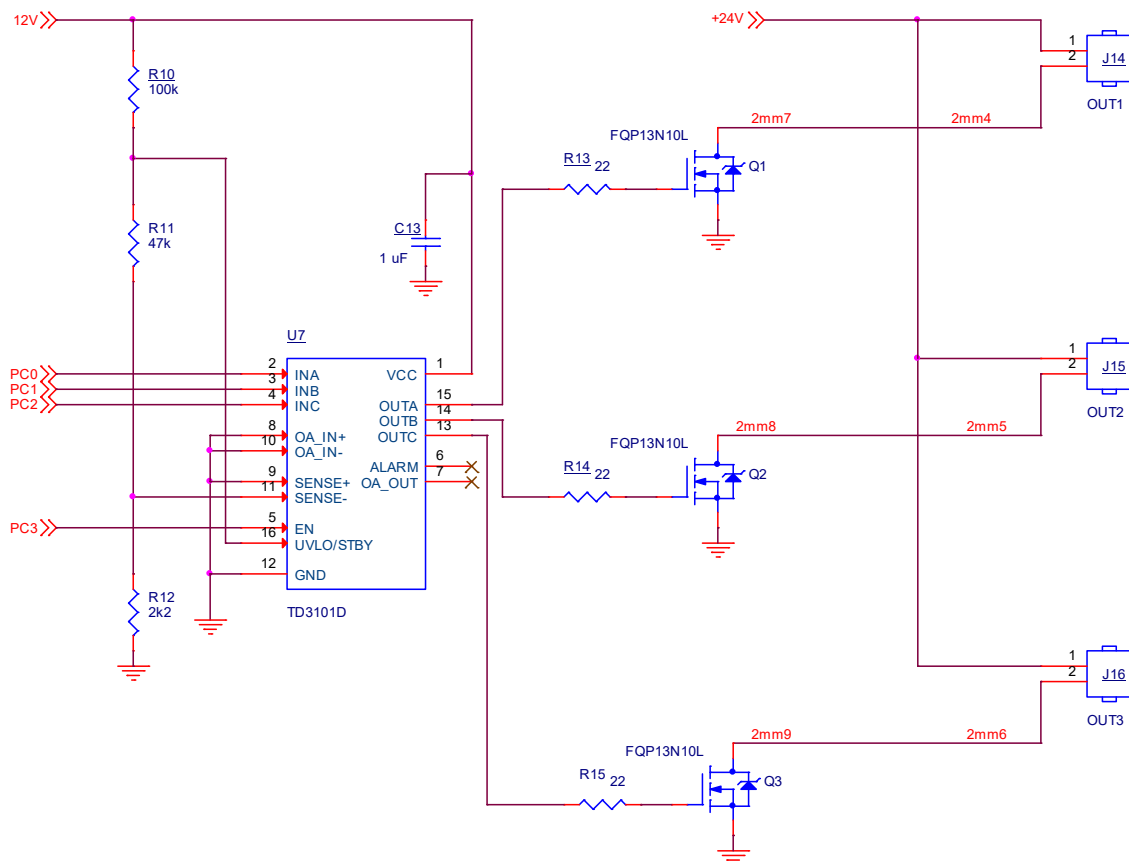


Fig.3. Power Drivers.

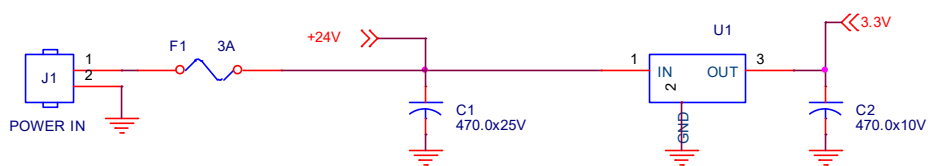


Fig.4.Alimentación.

Se usa un regulador del tipo Buck, para minimizar las pérdidas de potencia, dada la gran diferencia entre el voltaje de entrada y el de salida.

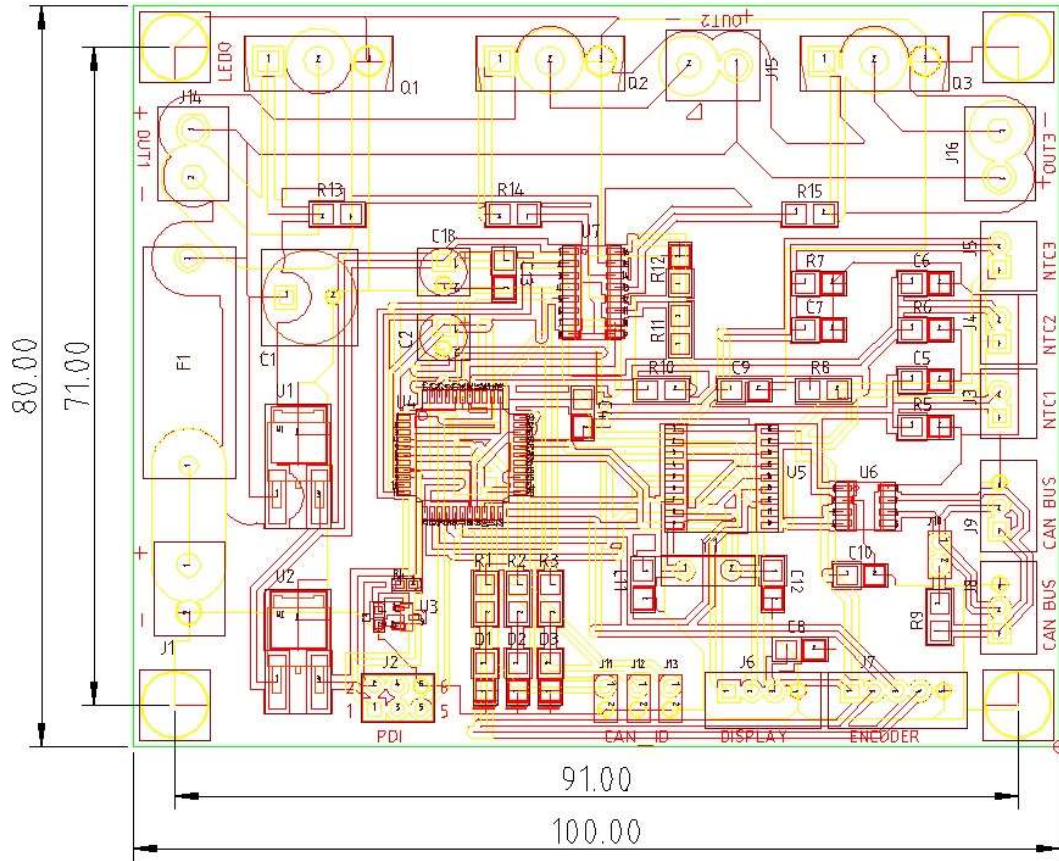


Fig.5. Dimensiones de la placa

REFERENCIA DE PROGRAMACION CAN BUS

Solo se procesan los mensajes dirigidos al Id del nodo, el cual puede modificarse con los jumpers presentes en la placa de acuerdo con la fórmula:

$$\text{CAN_ID} = 1536 + ((\text{PB2} \ll 2) | (\text{PB1} \ll 1) | \text{PB0});$$

Los jumpers actúan en la fórmula, cuando están abiertos.

COMANDOS DE LA PLACA CAN_COOLER_BOARD

COMANDO (BYTE0)	DATA BYTES	DESCRIPCION
1 SET KpA	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
2 SET KiA	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
3 SET KdA	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
4 ENABLE/DISABLE A	BYTE1	HIGH/LOW
5 SETPOINT A	BYTE1=LOW BYTE2=HIGH	°c X 100
6 SET KpB	BYTE1=LOW	ENTERO 16 BITS

	BYTE2=HIGH	
7 SET KiB	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
8 SET KdB	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
9 ENABLE/DISABLE B	BYTE1	HIGH/LOW
10 SETPOINT B	BYTE1=LOW BYTE2=HIGH	°c X 100
11 SET KpC	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
12 SET KiC	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
13 SET KdC	BYTE1=LOW BYTE2=HIGH	ENTERO 16 BITS
14 ENABLE/DISABLE C	BYTE1	HIGH/LOW
15 SETPOINT C	BYTE1=LOW BYTE2=HIGH	°c X 100
16 ENABLE ALL	NO	
17 DISABLE ALL	NO	
18 LEER TEMPERATURAS	NO	
19 LEER SETPOINTS	NO	

A todos los comandos desde el 1 al 17, el nodo responde con la siguiente estructura de datos:

Byte 0: Comando recibido

Byte 1: Parámetro modificado (Low byte)

Byte 2: Parámetro modificado (high byte)

Al commando 18 (Leer temperaturas), el nodo responde con la siguiente estructura de datos:

Byte[0] = 18

Byte[1] = TA_LOW

Byte[2] = TA_HIGH

Byte[3] = TB_LOW

Byte[4] = TB_HIGH

Byte[5] = TC_LOW

Byte[6] = TC_HIGH

La temperatura en °C la calculamos como:

$$T = (T_HIGH * 256 + T_LOW) / 100$$