Compact Laser Controller With Over Current Protection

Features:

- Easy to use, low power consumption, compact
- Faster response to faults
- Protects lasers from over current
- External fault indicator
- External reset button to reset the laser in normal condition
- Realtime displaying of current flowing

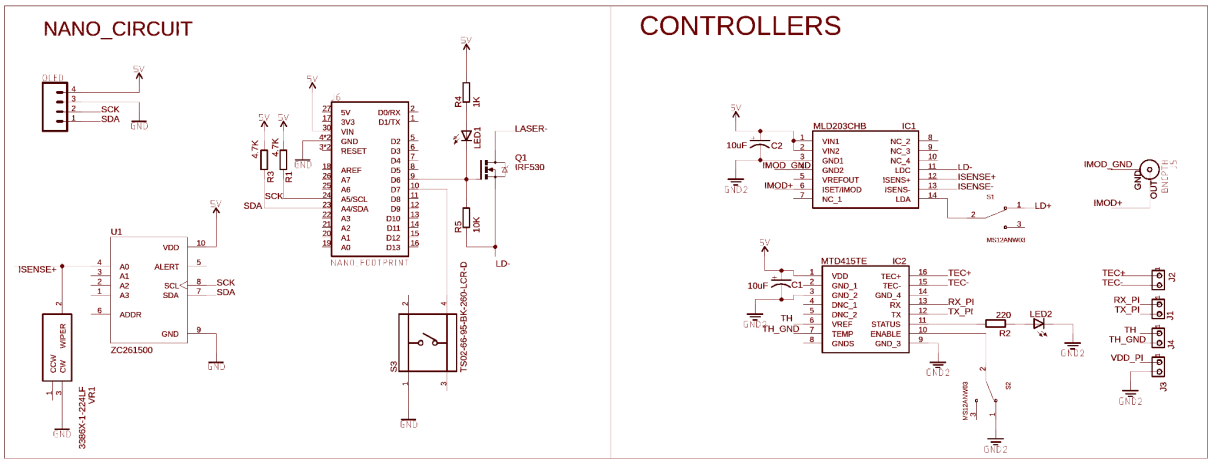
Applications:

- To protect the laser diodes from over current.
- To monitor the current flowing through the laser diodes
- To protect and monitor other loads/devices too.

Components Netlist:

- IRF530
- Arduino Nano
- ADS1115-16 bit ADC
- MLD203CHBE
- MTD415TE
- LEDs
- Potentiometer
- Push button
- Resistors
- Capacitors
- OLED display
- BNC connector
- Jumper connectors

Circuit Diagram:



Description:

- This circuit allows real time monitoring of the current flowing through the laser diode, It includes Arduino nano board to monitor the isense pins constantly. The circuit uses external ADC as the NANO's ADC is 10-bit and it does not provide enough resolution for our implementation. It uses ADS1115 to monitor Isense pins and transfer data over to NANO via I2C protocol.
- MLD203CHBE has an internal 1 Ohm resistor connected across Isense+ and Isense- pins so that the voltage across pins is directly related to the current flowing through the laser diode.
- The gate pin of the nMOS is driven by the digital pin of NANO. In over current condition the mosfet cuts off the laser diode from the circuit.
- The OLED display shows the real time reading of the current flowing through the laser diode.
- With the help of an external push button the circuit can be reset back to normal operation after current drops down to the normal level.

Pin Connections:

Arduino Nano:

- Pin 30 -> +5 volts
- Pin 4 -> ground (0 volts)
- Pin 23 -> I2C - SDA pin pulled high with 4.7Kohm resistor
- PIn 24 -> I2C - SCL pin pulled high with 4.7Kohm resistor
- (Why pullup resistors used? : In I2C protocol it is always recommended to use pull up resistors in order to define the active and inactive state of the I2C buses)
- Pin 9 -> Driving pin for Gate terminal of Mosfet
- Pin 10 -> Used to externally reset the circuit, to bring back the circuit to normal operation

Mosfet:

- Gate: Connected to D6
- Drain: Connected to the negative input terminal of the laser diode pin.
- Source: Connect to gate with 10k Ohm resistor and also connected to negative terminal (LD-) of laser controller.
- Cathode of Fault indicator LED is connected to the gate terminal and anode is connected to +Vcc via a resistor.

OLED display:

- SCK -> Connected to the A5 pin of nano
- SDA -> Connected to the A4 pin of nano
- Vcc -> +5 volts
- Gnd -> Ground

ADS1115 ADC:

- Pin 4 -> Connected to Isense- pin of MLD203CHBE, pulled down with 50K pot
- Pin 7 -> Connected to the A4 pin of nano

- Pin 8 -> Connected to the A5 pin of nano
- Pin 9 -> Ground
- Pin 10 -> +5 volts

MLD203CHBE (Constant Current Driver):

- Pin1 and Pin2 -> Connected together and to +5 volts
- Pin 3 ->Grounded, 10uF capacitor is connected in parallel to prevent voltage spikes
- Pin 4 -> Connected to Imod- pin used to modulate current to laser diode
- Pin 6 -> Connected to the +ive terminal of current modulator
- Pin 12 -> Connected to A0 pin of ADC
- Pin 13 -> Connected to ground
- Pin 14 -> Connected to Laser Diode's anode with switch in series
- Pin 11 -> Connected to the source terminal of the mosfet

MTD415TE (Laser TEC Driver):

- Pin 1 -> Connected to +5 volts
- Pin 2 -> Connected to ground and 10uF capacitor is connected to reduce voltage spikes
- Pin 6 -> Connected to the laser's TH pin
- Pin 7 -> Connected to the TH ground pin
- Pin 9 -> Grounded
- Pin 10 -> Connected to the switch enabled when pulled down
- Pin 11 -> Connect to the anode of the led to show the status
- Pin 12 -> TX pin to communicate with raspberry pi via UART protocol
- Pin 13 -> RX pin to communicate with raspberry pi via UART protocol
- Pin 15 -> Connected to the negative terminal of the laser's internal TEC
- Pin 16 -> Connected to the positive terminal of the laser's internal TEC

Technical Data:

Supply Voltage	4.5 to 5 V
Operating Temperature	-20 to +70 °C
Current Threshold Limit	0 - 120mA (programmable)
Current Reading accuracy	3.85% off in **Volt** range of modulation voltage
	4.76% off in **mVolt** range of modulation voltage
Example	When U_{ISET}[V] = 1V is applied according to data sheet I_{LD}[A] = 80mA but it reads 76-77mA
	When U_{ISET}[V] = is 400mV applied

	according to data sheet I_{LD} [A] = 32mA but it reads 26.5 - 27mA

For reference,

$$I_{LD}[A] = \frac{U_{ISET}[V]}{12.5[\Omega]}$$

Duration to read 1 Value	1.00 second
ADC Voltage reading range	± 6.144V
Conversion Factor	FSR/2^{16}= 12.288V/65,536 ≈ 0.1875 mV/bit
Mosfet $turn_{ON}$ delay time	9ns (typical) and 20ns (maximum)
Mosfet $turn_{OFF}$ delay time	26ns (typical) and 50ns (maximum)

Code:

```
#include <Wire.h>     // For I2C protocol
#include <Adafruit_ADS1X15.h>     // For ADS1115
#include <U8glib.h>     // For oled display
U8GLIB_SH1106_128X64 oled(U8G_I2C_OPT_NONE);      //Defining oled type
#define out_pin 6     // Defining pin 6 as gate driving pin
#define rst_pin 7     // Defining pin 7 as reset pin

const float V_th = 115.000;   // Defining cutoff value
const float FSR = 0.1875;     // Full Scale Range per bit for ADS1115
int16_t adc0;           // Global variable to hold ADC value
float voltage;          // Global variable to hold converted voltage value
Adafruit_ADS1115 ads;   // Create an ADS1115 object

void setup() {
  Serial.begin(115200);   // Start the Serial Monitor
  pinMode(out_pin, OUTPUT);   // assigning pin 6 as output pin
  pinMode(rst_pin, INPUT_PULLUP); // enabling internal pullup resistor of pin 7

  Serial.println("Initializing ADS1115...");    // print on serial monitor
  ads.begin();     // Initialize the ADS1115
}

void loop() {
  read_analog_value();      // Calling function to read analog value

  //enters only when voltage reading is greater than the cutoff.
  if(voltage >= V_th){
    while(digitalRead(rst_pin) != LOW){     //will stay in the while loop  until and unless external rst
button is pressed as on pin 7.
```

```
    digitalWrite(out_pin, LOW);       // will turn off the mosfet switch
    read_analog_value();
    serial_display();
    delay(1000);    //delay of 1 second
    }
  }
  else{
    digitalWrite(out_pin, HIGH);      // else it will keep mosfet on.
  }

  serial_display();     // Calling function to print on serial monitor
  delay(1000);    // Add a small delay
}

// Function to read analog value from ADS1115
void read_analog_value(){
  adc0 = ads.readADC_SingleEnded(0);     // Read the analog input from channel 0 (A0 pin)
  // Convert the ADC reading to voltage (assuming default gain)
  // ADS1115 has a resolution of 16 bits and a reference voltage of 6.144V
  voltage = adc0 * FSR;    // FSR = 12.288/65536 = 0.1875 = represents voltage of each bit
  oled_display(voltage);    // displaying on oled display
}

// Function to print on Serial Monitor
void serial_display(){
  Serial.print("ADC0: ");
  Serial.print(adc0);
  Serial.print("\tVoltage: ");
  Serial.println(voltage, 6);      // Print the result
}

// Function to print on OLED
void oled_display(float voltage){
  oled.firstPage();     // Begin the first page of the OLED display
  do {
    oled.setFont(u8g_font_profont12);     // Set the font for the display
    oled.setPrintPos(0, 15);      // Set the cursor position for the voltage label
    oled.print("Voltage (in mV):");     // Print the voltage label
    oled.setPrintPos(0, 25);
    oled.print(voltage, 4);     // Print the voltage value with 4 decimal places
    oled.setPrintPos(0, 45);
    oled.print("Current (in mA):");
    oled.setPrintPos(0, 55);
    oled.print(voltage, 4);
  } while (oled.nextPage() );
}
```