```
#include <Wire.h>
#define N_ELEMENTS 4

const int LedPin = 9;
const int LDRPin = A0;


struct arduino_info {
  int endereco;
  double ganho;
};

int address=0;
int n=0;
int j=0;
int found_elements=1;
arduino_info elements[N_ELEMENTS];
byte control=1;
unsigned long t_0,diff;
int recolhe_valores=0, calibre_count=0;
int PWM_Calibre=255;


void setup() {
  Serial.begin(9600);
  Serial.println("Begin Setup");
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(2,INPUT_PULLUP);
  pinMode(3,INPUT_PULLUP);
  pinMode(4,INPUT_PULLUP);
  pinMode(5,INPUT_PULLUP);
  pinMode(6,INPUT_PULLUP);
  pinMode(7,INPUT_PULLUP);
  pinMode(8,INPUT_PULLUP);

  digitalWrite(LED_BUILTIN,HIGH);
  delay(500);
  digitalWrite(LED_BUILTIN,LOW);


  // Various initialization processes needed
(such as reading adress pins, enabling general
call, and using builtin led for control purposes)

  for (int i=2; i <= 8; i++){
    bitWrite(address, n, !digitalRead(i));
    n++;
  }//reading the address coded in digital ports
from 2 to 8

  elements[0].endereco=address;  //Saving own
address
  Wire.begin(address);  //Initialize I2C
comunication
  bitSet(TWAR, TWGCE);  //Enable general call
  Wire.onReceive(receiveEvent);

  delay(100);

  propagate_address(address);
//Propagates the address so its know by everyone
  sort_copy(&elements[0],found_elements);
```

```
//Sort found addresses for use in calibration
  calibracao(&elements[0],found_elements);
//Calibration
  Serial.println("Setup ended");

}

void loop() {
  Serial.print(" \n");
  for (j=0;j<found_elements;j++){
    Serial.print("Endereco: ");
    Serial.print(elements[j].endereco);
    Serial.print("\t");
    Serial.print("Ganho: ");
    Serial.print(elements[j].ganho);
    Serial.print("\t");
    Serial.println(j);
  }

  delay(500);



}

void receiveEvent(int numBytes){
  String request="";

  while(Wire.available()){
    request += (char) Wire.read();
  }
  switch(request[0]){
    case 'E':
      elements[found_elements].endereco=
request[1];
      found_elements++;
      t_0=micros();
      break;

    case 'C':
      if (request[1]=='S'){
        recolhe_valores=1;
      }
      if (request[1]=='E'){
        recolhe_valores=0;
        calibre_count++;
      }
      break;
  }




}

void sort_copy (arduino_info* arr, int size){
  for (int k=0; k<(size-1);k++){
    for (int w=0; w<(size-(k+1));w++){
      if (arr[w].endereco > arr[w+1].endereco){
        arduino_info temp;
        temp.endereco= arr[w].endereco;
        arr[w].endereco=arr[w+1].endereco;
        arr[w+1].endereco=temp.endereco;
      }
    }
  }

}

void sampleADC(arduino_info* arr)
{
  double Value = 0;
  for (int k=0;k<10;k++)
  {
    Value += analogRead(LDRPin);
  }
  Value /= 10;
  arr[0].ganho=Value;
}

void calibracao (arduino_info* elements, int
found_elements){


  while (calibre_count<found_elements){
    if (elements[calibre_count].endereco ==
address){
      analogWrite(LedPin,PWM_Calibre);  //liga
led

      t_0=micros();//Waits to ensure everything
is read (se calhar considerar resposta para
garantir comunicação)
      diff=0;
      while(diff<1000000){
        diff=micros()-t_0;
      }

      while (control != 0){ //Transmite mensagem
para os outros colocarem a escuta
        Wire.beginTransmission(0);
        Wire.write("CS");
        control = Wire.endTransmission(true);
      }
      control=1;

      sampleADC(&elements[calibre_count]);  //Lê
valores

      t_0=micros();//Waits to ensure everything
is read (se calhar considerar resposta para
garantir comunicação)
      diff=0;
      while(diff<1000000){
        diff=micros()-t_0;
      }

      analogWrite(LedPin,0);//Desliga Led

      while (control != 0){ //Transmite mensagem
de flag para sairem de escuta
        Wire.beginTransmission(0);
        Wire.write("CE");
        control = Wire.endTransmission(true);
      }
      control=1;

      calibre_count++; //incrementa

    }
    else{
      if (recolhe_valores == 1){
        digitalWrite(LED_BUILTIN,HIGH);
        sampleADC(&elements[calibre_count]);
        //talvez apenas evocar uma vez, discutir
com o duarte ou seja apos o adc meter
recolhe_valores=0
      }
      digitalWrite(LED_BUILTIN,LOW);
    }

  }
  for (int i=0;i<found_elements;i++){

elements[i].ganho=transform_ADC_in_lux(elements[i
].ganho)/PWM_Calibre;
  }
}

void propagate_address(int address){
  while (control != 0){
    Wire.beginTransmission(0);
    Wire.write('E');
    Wire.write(address);
    control = Wire.endTransmission(true);
  }
  control=1;

  t_0=micros();
  while(diff<500000)
  {
    diff=micros()-t_0;
  }
}

double transform_ADC_in_lux(double sensorValue){
  double V_sensorValue=0, R_LDR=0, lux=0;
  double m = -1.2618595, b = 2.54480706;

  V_sensorValue = (5*sensorValue/1023);
  R_LDR = (10/V_sensorValue)*(5-V_sensorValue);
  lux = pow(10,b)*pow(R_LDR,m);
  return lux;

}
```