```
//    Author : Jose Goncalves
//    jose.braga.pt@gmail.com

//    Teacher : Nuno Peixoto
//    University : IPCA
//    17.03.2014

#include <stdio.h>

#include <sys/types.h>
#include <netinet/in.h>
#include <fcntl.h>
#include <stdlib.h>



//------------------------------------------- melhorias a fazer ---------------------------------------------------

// cria varios filedescriptor desnecessariamente ?  sempre que é chamada funão Temp() e HR.
// criar class


//-----------------------------------------------------------------------------------------------------------------
//------------------------------------------- I2C : Temperature ---------------------------------------------------
//-----------------------------------------------------------------------------------------------------------------




   char* Temp()

   {

    char Buffer[4], buff[4] ;

    float a , b , temp ;

    //  int fd = open("/dev/i2c-2", O_RDWR);
    //
    //  if (fd < 0) {
    //                 printf("Could not open memory\n");
    //              }
    //
    //  if (fd > 0) {
    //                printf("memory open successfully .\n");
    //              }

    // http://elinux.org/EBC_Exercise_05_I2C   i2ctools
    // http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
    // http://elinux.org/Interfacing_with_I2C_Devices
    // http://www.jumpnowtek.com/index.php?option=com_content&view=article&id=69&Itemid=78

  int fd1;
  char *filename = "/dev/i2c-2";
  if ((fd1 = open(filename, O_RDWR)) < 0) {
      /* ERROR HANDLING: you can check errno to see what went wrong */
      perror("Failed to open the i2c bus");
      exit(1);
  }

  int addr = 0x40;          // The I2C address of the device


  if (ioctl(fd1,0x0703,addr) < 0)
              {
              printf("Failed to acquire bus access and/or talk to slave.\n");
              /* ERROR HANDLING; you can check errno to see what went wrong */
              exit(1);
              }

    // leitura da temperatura

    Buffer[0] = 0xE3; // Command to read temperature : sensor SHT25 from Sensorion

    write (fd1,Buffer,1);
    // int written_bytes_i2c = write (fd1,Buffer,1);

    // int readed_bytes_i2c = read(fd1,buff,3);
    read(fd1,buff,3);

    a = (float) buff[0] ;
    b = (float) buff[1] ;
    temp = (((( a*256 ) + b )/(256*256))*175.72) - 46.85;
    // printf("Temperatura %.3f \n",temp);

    //char *ret = malloc(sizeof(char)*9);
    char ret[9] ;

    sprintf(ret,"%.2f",temp);     //  convert float into a char[]


    // acrescentar  "TEMP:" no inicio da string

    ret[5]=ret[0];
    ret[6]=ret[1];
    ret[7]=ret[2];
    ret[8]=ret[3];
    ret[9]=ret[4];

    ret[0]='T';
    ret[1]='E';
```

```
    ret[2]='M';
    ret[3]='P';
    ret[4]=':';

    // return a pointer to the first element of the character array.
    return ret ;

    }



//-------------------------------------------------------------------------------------------------------------------
//-------------------------------------------- Relative Humidity ----------------------------------------------------
//-------------------------------------------------------------------------------------------------------------------



char* HR()

      {

       char Buffer[4], buff[4] ;
       float c , d , HR ;

       //  int fd = open("/dev/i2c-2", O_RDWR);
       //
       //  if (fd < 0) {
       //                 printf("Could not open memory\n");
       //              }
       //
       //  if (fd > 0) {
       //                printf("memory open successfully .\n");
       //              }

     // http://elinux.org/EBC_Exercise_05_I2C   i2ctools
     // http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/
     // http://elinux.org/Interfacing_with_I2C_Devices
     // http://www.jumpnowtek.com/index.php?option=com_content&view=article&id=69&Itemid=78

     int fd1;
     char *filename = "/dev/i2c-2";
     if ((fd1 = open(filename, O_RDWR)) < 0) {
         /* ERROR HANDLING: you can check errno to see what went wrong */
         perror("Failed to open the i2c bus");
         exit(1);
     }

     int addr = 0x40;          // The I2C address of the device

     if (ioctl(fd1,0x0703,addr) < 0) {
         printf("Failed to acquire bus access and/or talk to slave.\n");
         /* ERROR HANDLING; you can check errno to see what went wrong */
         exit(1);
     }

        // leitura da humidade relativa

        Buffer[0] = 0xE5;   // // Command to read Relative Humidity : sensor SHT25 from Sensorion
        write (fd1,Buffer,1);
        //int written_bytes_i2c_HR = write (fd1,Buffer,1);
        read(fd1,buff,3);
        //int readed_bytes_i2c_HR = read(fd1,buff,3);

        c = (float) buff[0];
        d = (float) buff[1];
        HR = (((( c*256 ) + d )/(256*256))*125) - 6;
        // printf("Humidade relativa %.3f \n\n",HR);


        //char *ret = malloc(sizeof(char)*9);
        char ret[7] ;

        sprintf(ret,"%.2f",HR);     //  convert float into a char[]

        // acrescentar  "HR:" no inicio da string

        ret[3]=ret[0];
        ret[4]=ret[1];
        ret[5]=ret[2];
        ret[6]=ret[3];
        ret[7]=ret[4];

        ret[0]='H';
        ret[1]='R';
        ret[2]=':';

        return ret ;

         }
```