



TRABAJO DE LA ASIGNATURA DE INFORMÁTICA

DISEÑO DE UN PESO POR MEDIO DE UNA CÉLULA DE CARGA

COAUTORES:

ALEJANDRO PALOMO DE LA PEÑA

CRISTINA NAVARRO BADOS

CARLOS MENDOZA

1. INTRODUCCIÓN

En este proyecto se utiliza este instrumento para diseñar una pequeña balanza capaz de detectar pequeñas cantidades de masa (inferiores a un kilogramo preferiblemente)

2. OBJETIVOS

Tanto el programa diseñado como el montaje constituyen esta balanza, que tiene como objetivos principales traducir las diferencias de voltaje de la galga a mediciones de masa, comunicarse con el ordenador y almacenar esas medidas y datos para consulta o cálculo de fuerzas.

3. GUIA DE USO DE LOS PROGRAMAS

VISUAL

En Visual Studio hemos creado un programa capaz de obtener tanto la fuerza como el peso que ejerce un objeto una vez colocado en elacrílico.

```
float obtener_fuerza(Serial* Arduino, char* port);  
float obtener_peso(Serial* Arduino, char* port);  
int Enviar_y_Recibir(Serial* Arduino, const char* mensaje_enviar, char*  
mensaje_recibir);  
void crear_fichero_txt(char* mensaje);
```

Para realizarlo hemos creado una serie de funciones que explico a continuación:

- 1) Esta función devuelve un float pues el resultado dispondrá de decimales. Por otro lado le pasaremos los valores del Arduino y el puerto al que está conectado. Al recibir el mensaje de Arduino, este lo disponemos en formato char pero necesitamos convertirlo a float por lo que creamos un switch case el cual, mientras recorremos la cadena, transforma la cadena en un numero

```
float obtener_fuerza(Serial* Arduino, char* port) {  
  
    float peso = 0;  
  
    int fuerza;  
  
    int bytesRecibidos;  
  
    char mensaje_recibido[MAX_BUFFER];  
    float numero = 0;  
    int i, divisor = 10, estado = 0;  
  
    if (Arduino->IsConnected())
```

```

{
    bytesRecibidos = Enviar_y_Recibir(Arduino, "GET_PESO\n",
mensaje_recibido);

    if (bytesRecibidos <= 0)
    {
        printf("\nNo se ha recibido respuesta a la petición\n");
    }
    else
    {
        for (i = 0; mensaje_recibido[i] != '\0' && estado != 3 &&
i < MAX_BUFFER; i++)
            switch (estado)
            {
                case 0: // Antes del número
                    if (mensaje_recibido[i] >= '0' &&
mensaje_recibido[i] <= '9')
                    {
                        numero = mensaje_recibido[i] - '0';
                        estado = 1;
                    }
                    break;
                case 1: // Durante el número
                    if (mensaje_recibido[i] >= '0' &&
mensaje_recibido[i] <= '9')
                        numero = numero * 10 +
mensaje_recibido[i] - '0';
                    else
                        if (mensaje_recibido[i] == '.' ||
mensaje_recibido[i] == ',')
                            estado = 2;
                        else
                            estado = 3;
                    break;
                case 2: // Parte decimal
                    if (mensaje_recibido[i] >= '0' &&
mensaje_recibido[i] <= '9')
                    {
                        numero = numero +
(float)(mensaje_recibido[i] - '0') / divisor;
                        divisor *= 10;
                    }
                    else
                        estado = 3;
                    break;
            }

            printf("\n FUERZA = %.2f N\n", numero*9.81);
        }
    }
    else
    {
        printf("\nNo se ha podido conectar con Arduino.\n");
        printf("Revise la conexión, el puerto %s y desactive el monitor
serie del IDE de Arduino.\n", port);
    }
}

```

```

    return peso;
}

```

- 2) La opción 2 sería calcular el peso del objeto. Este peso lo dispondremos en formato float.

Mediante la función Enviar_y_Recibir pasaremos el valor GET_PESO conocido en Arduino a Visual

```

float obtener_peso(Serial* Arduino, char* port)
{
    float peso = 0;
    int bytesRecibidos;
    char mensaje_recibido[MAX_BUFFER];

    if (Arduino->IsConnected())
    {
        bytesRecibidos = Enviar_y_Recibir(Arduino, "GET_PESO\n",
mensaje_recibido);
        if (bytesRecibidos <= 0)
        {
            printf("\nNo se ha recibido respuesta a la petición\n");
        }
        else
        {
            printf("\nPESO = %s kg\n", mensaje_recibido);
            crear_fichero_txt(mensaje_recibido);
        }
    }
    else
    {
        printf("\nNo se ha podido conectar con Arduino.\n");
        printf("Revise la conexión, el puerto %s y desactive el monitor
serie del IDE de Arduino.\n", port);
    }

    return peso;
}

```

- 3) Por último, se creará un fichero con todos los pesos realizados para que si en algún momento necesita recordar el peso de un objeto se encuentre almacenado en un fichero llamado Pesos.txt:

```

void crear_fichero_txt(char* mensaje) {

    FILE* fichero;
    int i;
    errno_t e;

```

```

e = fopen_s(&fichero, "Pesos.txt", "a");

if (fichero == NULL)
    printf("No se ha podido guardar los datos\n");
else {

    fprintf(fichero, "%s\n", mensaje);

    fclose(fichero);
}
}

```

Todas estas funciones serán introducidas en el main, en el switch-case creado según la opción que le convenga. En la opción 3 podemos leer el fichero creado para recordar nuestras mediciones anteriores.

*El valor **ctrl** permite leer el fichero de tal forma que, al ir introducido en un do-while (en el que introducimos la cantidad de palabras que hay en una línea del fichero) lo igualamos a un fscanf y guardamos el peso en el contenido de mens

```

int main(void)
{
    Serial* Arduino;
    char puerto[] = "COM3"; // Puerto serie al que está conectado Arduino
    int opcion_menu;
    float peso;
    float tara;
    setlocale(LC_ALL, "es-ES");
    FILE* fichero;
    errno_t e;
    int ctrl;
    Arduino = new Serial((char*)puerto);
    char mens[N];

    do
    {
        opcion_menu = menu();

        switch (opcion_menu)
        {
            case 1:

                peso = obtener_fuerza(Arduino, puerto);

```

```

        break;
    case 2:
        peso = obtener_peso(Arduino, puerto);

        break;
    case 3:

        e = fopen_s(&fichero, "Pesos.txt", "rt");

        if (fichero == NULL) {
            printf("Error al leer el fichero");
        }
        else {

            do {

                ctrl = fscanf_s(fichero, "%s", mens, N);

                if (ctrl == 1) {
                    printf("Peso = %skg\n", mens);
                }

            } while (ctrl == 1);
            fclose(fichero);
        }

        break;
    case 4:
        break;
    default: printf("\nOpción incorrecta\n");
}
} while (opcion_menu != 4);

return 0;
}

```

ARDUINO

```

#include <LiquidCrystal_I2C.h>
#include "HX711.h"

String mensaje_entrada;
const int DOUT=A1;
const int CLK=A0;
HX711 balanza;
LiquidCrystal_I2C lcd(0x27,16,2);
const int inputPin =8;
float peso;

```

Introducimos estas dos librerías necesarias para que funciones tanto la pantalla LCD como la placa ADC hx711

```
void
setup()
{
    Serial.begin(9600);
    pinMode(inputPin, INPUT);

    balanza.begin(DOUT, CLK);

    //Serial.print("Lectura del valor del ADC: ");
    //Serial.println(balanza.read());
    //Serial.println("No ponga ningún objeto sobre la balanza");
    //Serial.println("Destarando...");
    //Serial.println("...");

    balanza.set_scale(-515570.45); //La escala por defecto es 1
    balanza.tare(20); //El peso actual es considerado Tara.
    //Serial.println("Listo para pesar:");

    lcd.init();

    //Encender la luz de fondo.
    lcd.backlight();
    lcd.clear();
    //Ajustamos el cursor del LCD
    lcd.setCursor(0,0);
    // Escribimos el Mensaje en el LCD.

}
```

En el setup centramos la pantalla LCD para que el cursos salga desde arriba a la izquierda.

Por otro lado debemos marcar una escala (-515570.45)

```
void
loop()
```

```

{

if (Serial.available()>0)
{
    mensaje_entrada= Serial.readStringUntil('\n');
    if (mensaje_entrada.compareTo("GET_PESO")==0)
    {
        peso=digitalRead(inputPin);
        peso=balanza.get_units(20),3;
        Serial.println(peso);
        lcd.print(peso);

    }
}
lcd.setCursor(0,0);
lcd.print(' ');

}

```

En el loop pasamos el mensaje de GET_PESO a visual para que podamos trabajar con el desde allí y lo imprimimos en la pantalla LCD

4. HARDWARE

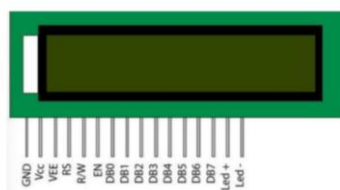
PLACA LCD

Una pantalla **LCD** son dispositivos diseñados para mostrar información en forma gráfica. **LCD** significa Liquid Crystal Display (Display de cristal líquido). La mayoría de las pantallas **LCD** vienen unidas a una **placa** de circuito y poseen pines de entrada/salida de datos.



Es bastante sencillo unir la pantalla a Arduino mediante una librería llamada LiquidCrystal que viene junto con Arduino IDE.

Nuestra pantalla consta de 2 filas por 16 columnas, por lo que podemos imprimir caracteres en 2 filas las cuales pueden llegar hasta 16 caracteres.



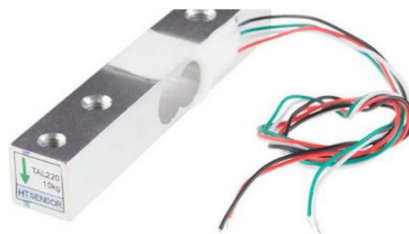
Constan de 16 pines:

- Pin 1 – VSS o GND
- Pin 2 – VDD o alimentación (+5V)
- Pin 3 – Voltaje de contraste. Se conecta a un potenciómetro.
- Pin 4 – Selección de registro. Aquí se selecciona el dispositivo para su uso.
- Pin 5 – Lectura/Escritura. Dependiendo del estado (HIGH o LOW), se podrá escribir o leer datos en el LCD
- Pin 6 – Enable. Es el pin que habilita o deshabilita el LCD.
- Pin 7 hasta Pin 14 – Son los pines de datos por donde se envía o recibe información.
- Pin 15 – El ánodo del LED de iluminación de fondo (+5v).
- Pin 16 – El cátodo del LED de iluminación de fondo (GND).

La manera de conectarlo a Arduino lo muestro en el mapeo del proyecto

CELDA DE CARGA Y AMPLIFICADOR Hx711

Una **celda de carga** es particularmente rígida, tienen valores de resonancia muy buenos y tienden a tener ciclos de vida largos en la aplicación. Un sensor de carga o pesaje es un medidor de deformación interna, el rango de medición máxima de este producto es de 1kg de pesaje.



El **HX711** es un amplificador de célula de carga que te permite medir la fuerza (peso) con un Arduino, un ESP8266 o cualquier otra placa microcontroladora.



Sabiendo cuánto pesa por ejemplo una bombona vacía, fácilmente se puede controlar el estado de la bombona utilizando células de carga y el amplificador HX711 conectado a un Arduino.

La Celda de Carga de 1kg y Amplificador Hx711 son dispositivos que trabajan en conjunto para poder medir el peso o la fuerza externa, ya que la celda es un transductor que convierte el desplazamiento o deformación en señales eléctricas en milivoltios por lo cual se requiere el uso del Amplificador HX711 para poder tener la información digital y procesarla en alguna tarjeta de desarrollo Arduino, como es nuestro caso.

PLACA ARDUINO UNO

Se utilizará el Arduino Uno R3, la de uso generalizado para este tipo de proyectos. Es una placa robusta, dispone de 14 pines digitales de entrada y salida, 6 entradas analógicas, un oscilador de cristal de 16 MHz, un puerto usb, conectores de alimentación e ISCP y un botón reset.



Mediante el entorno de desarrollo Arduino IDE, se puede programar la placa en C o C++ mediante el desarrollo del programa, su compilación y su subida a la placa.

El Arduino se conectará con el HX711 y con el ordenador, para captar, procesar y enviar datos y valores, siendo un pilar para el funcionamiento del conjunto.

