



UNIVERSIDAD TÉCNICA NACIONAL
INGENIERÍA ELECTRÓNICA

Proyecto Final

Angie Marchena Mondell - 604650904

Programación I

Agosto 2021

Indice

Contenido

Indice	2
Contexto	3
Herramientas.....	3
Hardware	3
Arduino UNO	3
LCD	3
Componentes	3
Software.....	4
Arduino IDE	4
DevC++	4
SerialLib	4
Tutorial	4
Arduino	4
C++	5
Manual de usuario	6
Arduino	6
C++	6
Resultados.....	7
Lecciones aprendidas.....	9
Conclusiones.....	9
Referencias	9

Contexto

En la actualidad es muy común la experimentación con circuitos digitales, los cuales son muy usados para la creación de sistemas complejos computacionales, como a nivel de computadoras, sistemas integrados, SoC, o PIC, estos usualmente trabajan con voltajes pequeños, aceptables, entre 2 y 4 voltios, además de esto sabemos que un incremento de voltaje para estos circuitos puede ser fatal, por lo que es bueno saber con qué voltaje estamos tratando.

La idea principal es poder medir voltaje, con que alimentan a circuitos electrónicos, así como leds que indique que valores de voltaje son aceptados o que tan peligroso es para el circuito, además de poder ver en tiempo real desde el programa en C++ el voltaje marcado.

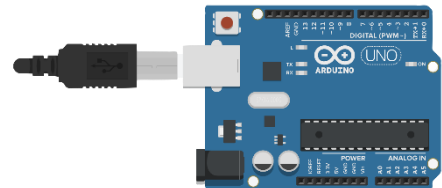
Herramientas

Para el desarrollo del sistema son necesario las siguientes herramientas:

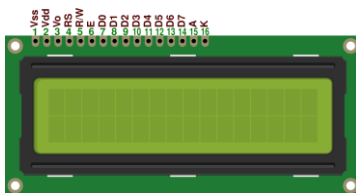
Hardware

Arduino UNO

Es el dispositivo encargado de realizar lo deseado en el proyecto, es el dispositivo electrónico programable.



LCD



Es un dispositivo electrónico el cual es utilizado para imprimir caracteres y datos los cuales deseamos que sean mostrado, es una pantalla de 16x2 dimensiones.

Componentes

- Variedad de componentes como resistencias, leds y protoboard.

Software

Arduino IDE



Este es el entorno de desarrollo integrado de Arduino, el cual está escrito en el lenguaje JAVA, es un programa libre el cual se puede descargar en el siguiente enlace.

- [Arduino Software](#)

DevC++

Este es el entorno de desarrollo integrado para el lenguaje de programación C/C++, su compilador es GCC, este se encarga de compilar y ejecutar aplicaciones en C++.



- [DevC++](#)

SerialLib



Es una biblioteca especializada desde C++ utilizada para comunicarse con el puerto serial del sistema Arduino, para así enviar datos importantes al mismo.

- [BibliotecaSerial](#)

Tutorial

Arduino

Para el correcto uso del sistema es necesario instalar todos los complementos de software anteriormente mencionados.

Como primer paso es identificar el puerto utilizado por el Arduino para establecer su comunicación serial, esto se puede ver en la siguiente imagen:

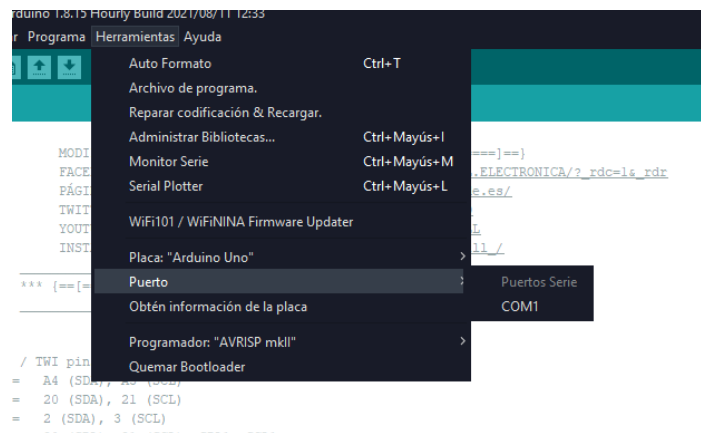


Figura 1: Puerto de comunicación con Arduino.

En este caso el Arduino esta usando el puerto COM, pero para el caso del usuario puede variar y debe verificar en el sistema de esta manera para ver de cual se trata.

Luego de esto en el programa debemos utilizar el puerto serial, esto se hace con el siguiente código:

```
// Para almacenar datos
char datos;

void setup() {
    // escucha el puerto serial
    Serial.begin(9600);
}

void loop() {

    // verifica si existe mensaje en el serial
    if(Serial.available() > 0)
    {
        // toma el valor del dato serial
        datos = Serial.read();
    }
}
```

Como vemos es muy sencillo para Arduino escuchar el puerto serial, y así tomar sus datos, los cuales son cadenas.

C++

Para el programa en C++, desde DevC++ se debe importar la biblioteca SerialClass, incluyendo en la carpeta, esto añadiendo SerialClass.cpp y SerialClass.h.

Luego de esto incluyendo en el código del main del programa de la siguiente manera:

```
#include "SerialClass.h"
```

Se puede ver en la siguiente imagen.

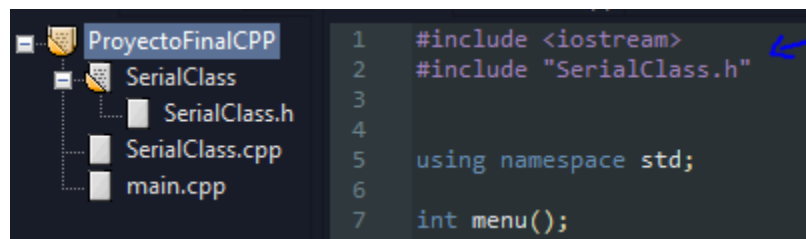


Figura 2: Desglose del programa en C++.

Con esto ya podemos usar la herramienta de manera correcta.

Luego es necesario definir nuestro serial, se hace de la siguiente manera:

```
Serial* Arduino = new Serial("COM1");
```

En ese caso usamos COM1, pero varia con el sistema.

Con esto tenemos listo para enviar datos desde C++.

Esto se hace con el siguiente código:

Acá verificamos que si esta conectado el serial, si es así mandamos el dato como se vio, donde la memoria es la longitud de un char, y luego el dato.

```
// instancia de puerto serial
Serial* Arduino = new Serial("COM6");
```

Figura 4: Código de comunicación serial.

Resultados

Acá podemos ver los resultados, se puede observar que el sistema a un voltaje menor que 2 V puede tener problemas para funcionar por eso se enciende la luz amarilla lo cual indica el voltaje medido, así como en la pantalla el valor del mismo.

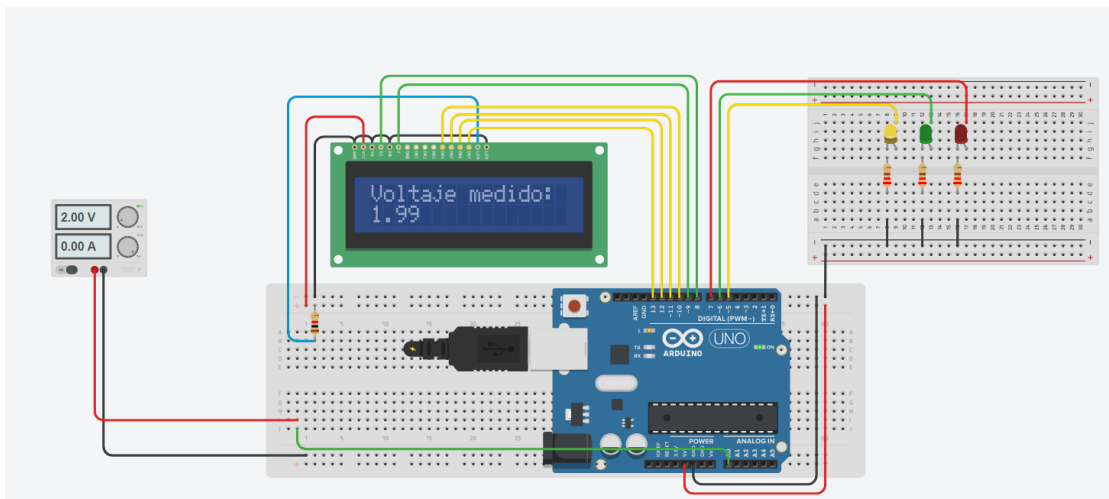


Figura 5: Sistema funcionando voltaje bajo.

En cambio en la figura 6 vemos un circuito con un voltaje controlable y aceptable para sistemas de electrónica digital.

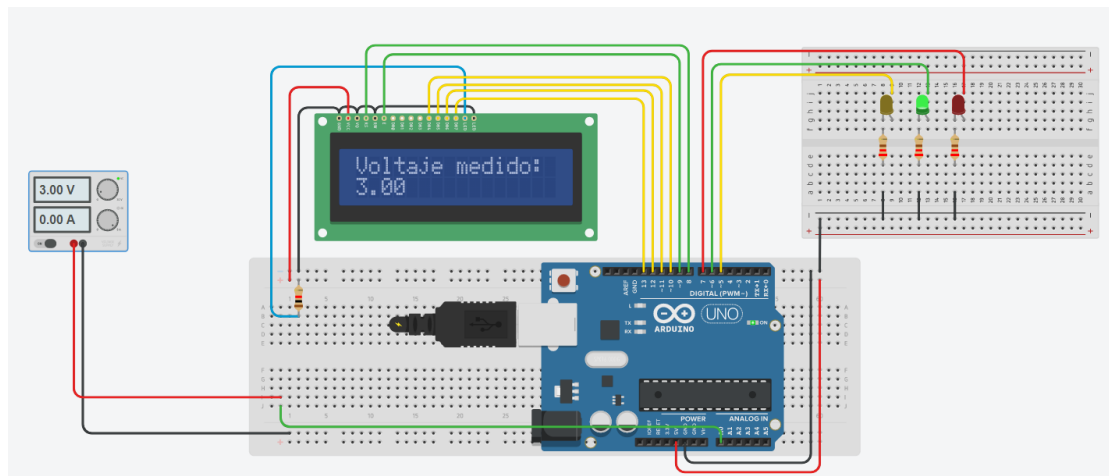


Figura 6: Sistema funcionando voltaje medio (ideal).

Por último la figura 7 vemos un voltaje un poco mas alto, con riesgo a sobrepasar los valores permitidos por circuitos eléctricos de microelectrónica como amplificadores o compuertas lógicas, además se ve el led indicando peligro en color rojo.

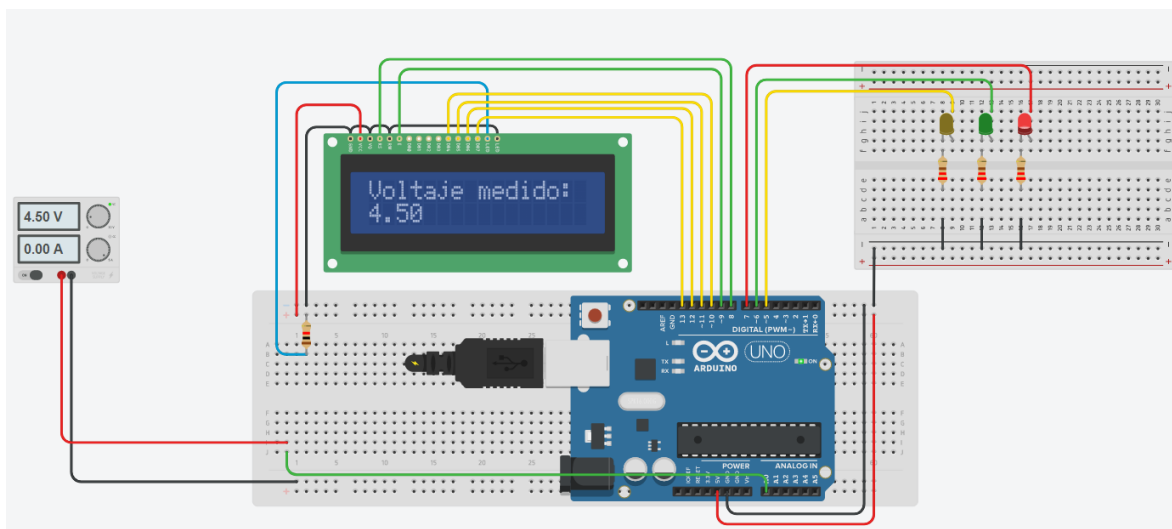


Figura 7: Sistema funcionando voltaje alto (limite).

El menú del sistema es el siguiente:

```

-----Menu-----
-- 1 - Medir voltaje
-- 2 - Apagar sistema
-- 3 - Ver datos en el puerto
-- 4 - Salir
-----Menu-----
-- Que desea realizar?
3

      Ha elegido la opcion Ver datos en el puerto
Voltaje actual: 4.50
  
```

Figura 8: Menú del sistema

Acá podemos ver que al presionar 1 nos envía la señal para que se realice la medición, con 2 apagamos el sistema y al final 3 vemos los datos del seria.

En la imagen se nos muestra esa opción, al presionar 3 nos envía el voltaje medido en ese instante de tiempo.

Lecciones aprendidas

Se aprendió con bastante firmeza que la comunicación del sistema con Arduino a nuestro nivel se hace mediante puerto serial, ya que es fácil de utilizar y que en general es de fácil acceso desde un mismo computador.

En Python también se puede utilizar estas funciones ya que se estudio ejemplos donde se utilizó Python.

El sistema sirve mientras este conectado a la computadora ya que los datos son mediante cables.

Si se desea de manera inalámbrica, debería de hacerse mediante módulo de wifi y hacer envíos de diferente manera, ya que serial no seria valido.

Conclusiones

C++ es un lenguaje fácil para utilizar para comunicación serial con Arduino, además de que existen muchas librerías para utilizar el puerto serial, por lo que habrá muchas maneras distintas.

Para el uso del LCD se puede utilizar un modulo I2C que relaciona las 16 entradas del LCD únicamente con 4, para mayor facilidad.

El Arduino es un sistema electrónico bastante potente y fácil de programar.

Referencias

A. (2019, 19 octubre). *Pantalla alfanumérica LCD 16X2 con Arduino*. Geek Factory.

<https://www.geekfactory.mx/tutoriales/tutoriales-arduino/pantalla-lcd-16x2-con-arduino/>

Theory, G. (s. f.). *Cómo usar Arduino y C++*. Geeky Theory. Recuperado 21 de agosto de 2021, de <https://geekytheory.com/como-usar-arduino-y-cplusplus>

Anexos

Fotos del sistema físico:

