

Mini Editor de Texto

*Seminario de Introducción a la
Microelectrónica y Programación de
Microcontroladores con Arduino*

Universidad Nacional de Quilmes,
TPI.

Mieres Federico, Ruiz Diaz Christian, Campos Diego, Bouix Cristian.

Indice

- *Materiales**
- *Sobre conexiones PS2**
- *Sobre conexiones LCD**
- *Sobre nuestra programación**
- *Circuito final**
- *Conclusiones.**

***Materiales**

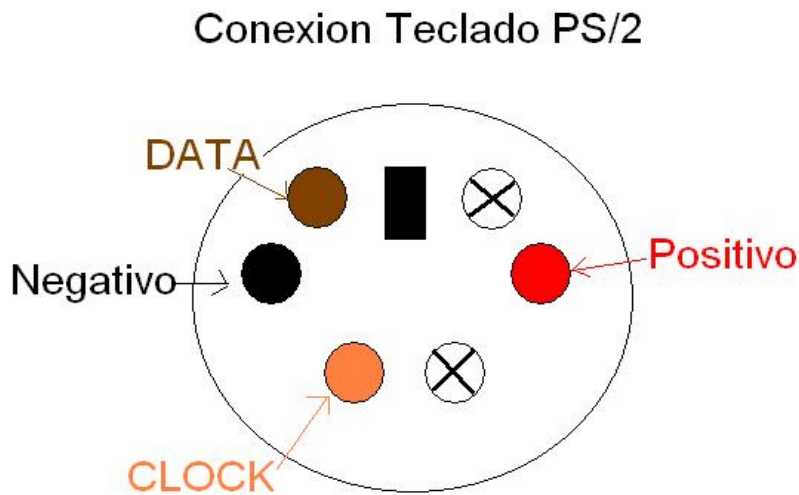
Pantalla LCD 16x2
2 conectores PS2 hembra
Teclado PS2

***Sobre conexiones PS2**

Tomamos la decisión de puentear los dos conectores PS2 entre ellos para emprolijar las conexiones. De esta manera un conector PS2 queda ligado al arduino y otro queda libre para conectar cualquier teclado PS2. Una vez las soldaduras fueron hechas, las verificamos con un tester.

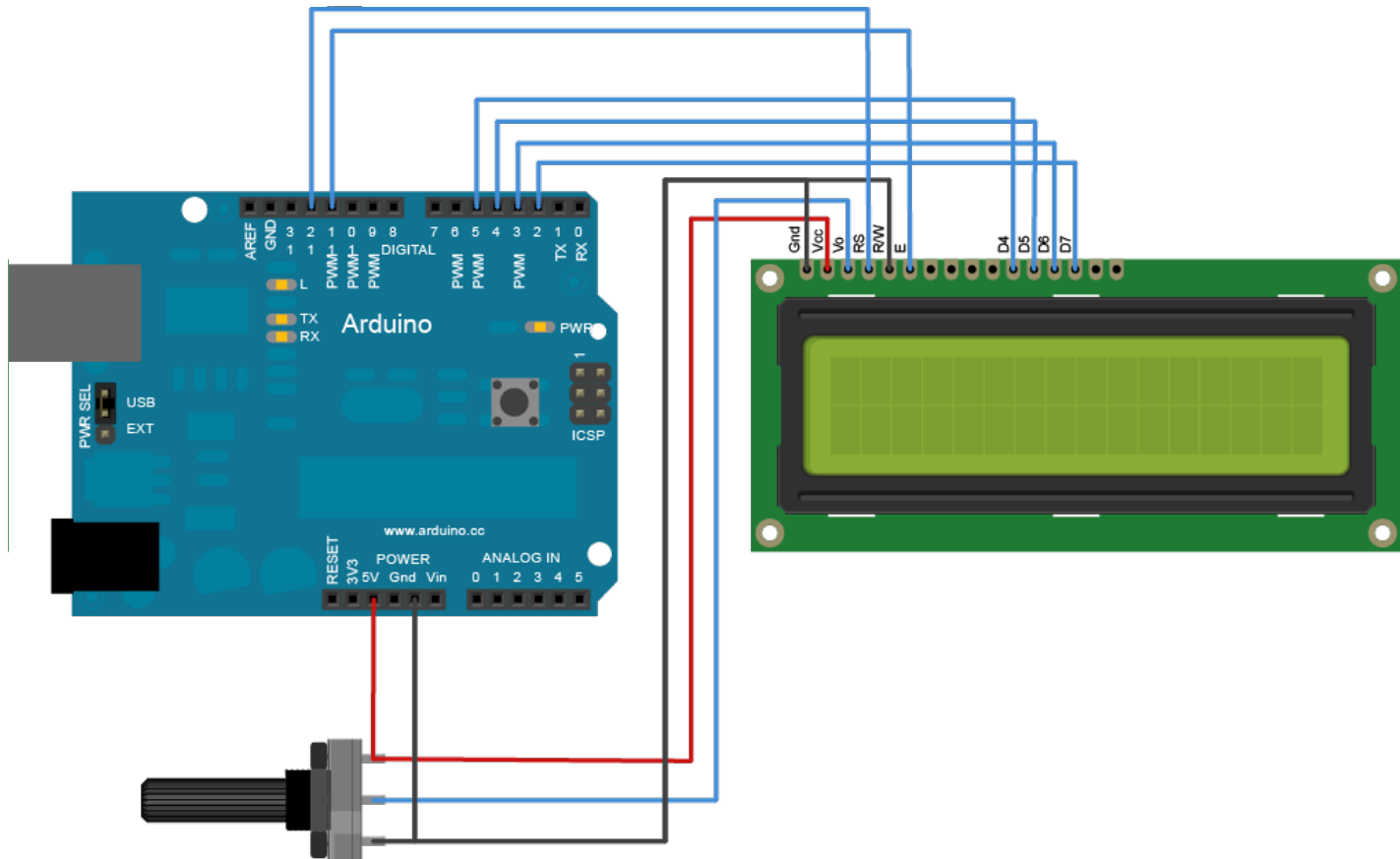
Lo siguiente que hicimos fue buscar la manera de realizar la conexión con el arduino. Aunque con dificultades para entenderlo, seguimos el esquema de la página de arduino: <http://www.arduino.cc/playground/Main/PS2Keyboard>. Después de varios intentos, y un teclado quemado en el camino, logramos la conexión definitiva.

Aquí mostramos una imagen de las conexiones correspondientes para el PS2 hembra:



***Sobre conexiones LCD**

Una vez hecha la conexión del teclado con el arduino procedimos a conectar la pantalla LCD al mismo, basandonos en el siguiente esquema(<http://arduino.cc/en/Tutorial/LiquidCrystal>) :



***Sobre nuestra programación**

Antes de poder dar comportamiento a nuestro arduino, teníamos que asegurarnos que seríamos capaces de reconocer lo que el teclado enviaba al arduino. Entonces probamos un código básico sacado de la misma página, pero tuvimos problemas. Así que seguimos buscando y dimos con:

```
#define CLK 8
#define DAT 9
void loop(){
  byte b = readkey();
  if (b == 0xF0) {
    // key released
    b = readkey(); }
  else { Serial.print(b); }
}
```

```

byte readkey(){
  byte _start = 0;
  byte buf = 0;
  byte _parity = 0;
  byte _stop = 0;
  waitClockLow();
  _start = digitalRead(DAT);
  if (_start == 0) {
    waitClockLow();
    for (byte c=0; c<8; c++){
      buf = buf | (digitalRead(DAT) << c);
      waitClockLow();
    }
    _parity = digitalRead(DAT);
    waitClockLow();
    _stop = digitalRead(DAT);
  }
  return buf;
}

void waitClockLow(){
  if (digitalRead(CLK) == LOW)
    while (digitalRead(CLK) == LOW){}
  while (digitalRead(CLK) == HIGH){}
}

```

Apartir de ese fragmento de código y a base de tocar una tecla y anotar lo impreso en el puerto serial, escribimos los binding que más tarde extrajimos en una librería. Una vez que terminamos con la función convertir que dado el byte retorna el character correspondiente, pasamos a modelar las funciones básicas de sincronización de teclado con el LCD.

También armamos una librería con este modelado. De esta forma, en un sketch de arduino tenemos solamente el código `#include <LocoLCDKeyboard.h>`

```

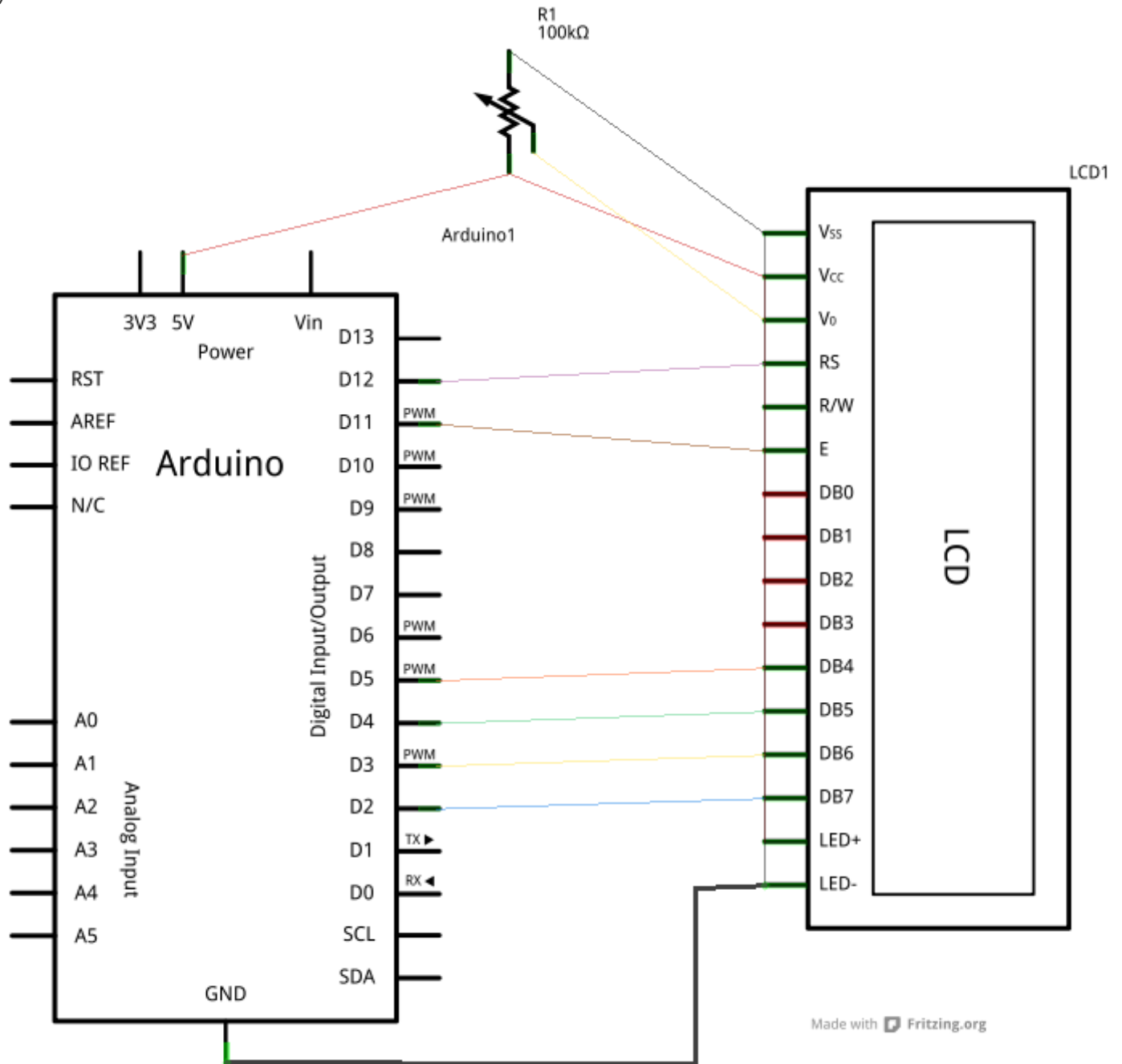
#include <LocoKeyboard.h> // librería bindings del teclado
#include <LocoLCDKeyboard.h> // librería modelado del editor de textos
#include <LiquidCrystal.h>
using namespace std;
#define LED 13
#define CLK 8
#define DAT 9
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
LLCDK l(CLK,DAT); // objeto representativo del editor
void setup(){
  pinMode(LED, OUTPUT);
  delayMicroseconds(50);
  lcd.begin(16, 2);
  Serial.begin(9600);
  lcd.cursor();
}
void loop(){
  int b = l.readkey();
  if (b == 0xF0) b = l.readkey();
  else {
    if ( b == TAG ) l.tagAct(lcd);
    else {

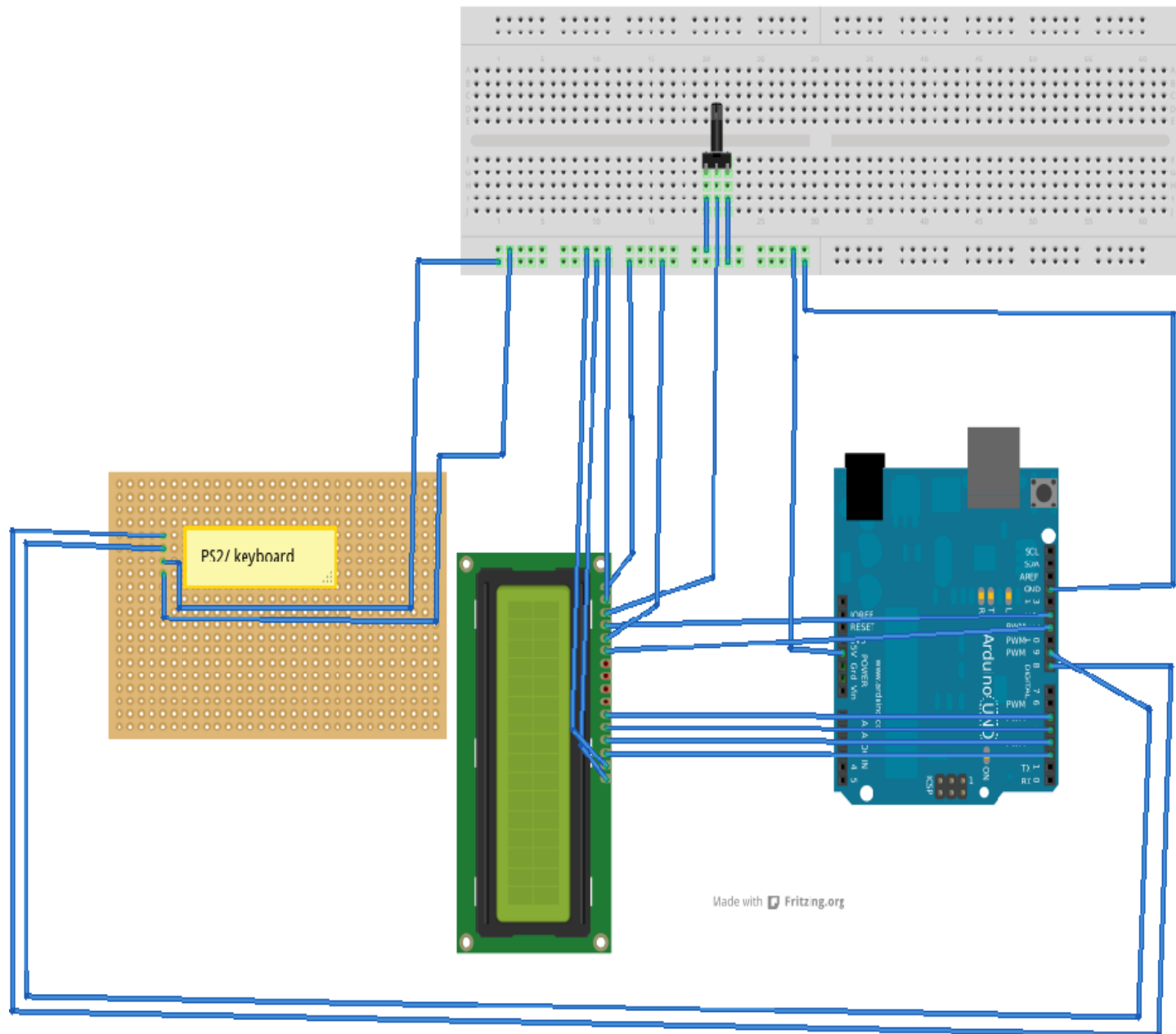
```

```
if (b == ESC) l.clean(lcd);
else {
    if (l.isMode(b)) l.setMode(b);
    else {
        if (isMove(b)) l.moveCursor(lcd,b);
        else{
            l.write(lcd, l.convert(b));
        }
    }
}
}}
```

***Circuito Final**

Finalmente diagramamos los circuitos que armamos utilizando Fritzing (<http://fritzing.org/>):





*Conclusiones

A nivel general más allá del Trabajo práctico final, la experiencia de trabajar con un arduino fue muy buena. A lo largo de la materia tuvimos un contacto cercano con el hardware y el código que desarrollamos fue a bajo nivel. vimos problemas de hardware, conexiones, malas soldaduras!, etc. Problemas a los cuales no estamos acostumbrados, siempre trabajamos en entornos donde el hardware no es un problema. La experiencia de trabajar con un arduino es totalmente grata.

Hablemos del Trabajo en sí; la búsqueda de librerías que faciliten nuestra labor, no cumplieron las expectativas, no obstante en lugar de restar, sumó; ¿ Por que sumó? Debido a que las

librerías que encontramos no funcionaban, nos dio lugar a desarrollar una propia, que no tiene el alcance de las que en teoría funcionan, pero funciona. Pudimos utilizar algunas ideas de las librerías, no obstante requirió trabajo.

El uso de hardware diverso, fue algo muy productivo, nunca habíamos usado un LCD, ni otros componentes que usamos a lo largo de la materia. Terminamos contentos con el producto final, un mini editor de texto, quedó trabajo para futuro, pero ahora hay una base. Puede escalar, o al menos brindar buenas ideas y se tienen varios problemas solucionados.

Nos quedamos con la librería funcionando, sencilla, pero se puede extender. El editor de texto funciona :D.

Así que, si se quiere hacer algo y no está todo disponible, igual se puede hacer :), y se aprende más.