****

**PROYECTO**

**CARPETA DE CAMPO**

Integrantes del grupo:

* Jonas Pereyra
* Máximo Cabral,
* Juan Martin Romero
* Mateo Shinocca
* Luciano Ojeda

Profesor: Gonzalo Consorti

Alumno: Máximo Cabral

Curso: 4°1 T.T

Proyecto: Mini instrumento electro-estático (touch)

Herramientas utilizadas en el proyecto

* 1x Arduino Uno R3
* 1x USB para programar
* 1x Protoboard
* 7x Sensor Touch
* 1x DF Player Mini HW-247A
* 84 MILLONES DE CABLES
* Rotary Encoder

Viernes 25/10/2024

Introducción al proyecto:

Una vez elegido el proyecto del mini instrumento electro-estático,decimos tras una serie de votos grupal que el instrumento a realizar sea un piano electro-estático

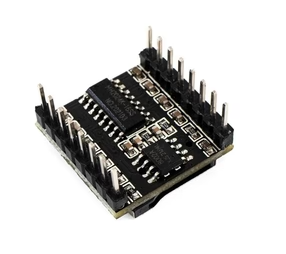
Nuestra idea principal del piano era hacerlo con botones y un buzzer, pero el profesor Gonzalo Consorti nos dijo que lo hagamos con sensores touch

Primero que nada empezamos a investigar sobre los siguientes componentes y su modo de uso, el cual desconociamos totalmente:

* DF Player Mini HW-247A
* Sensor Touch
* TRRS Audio Jack
* Rotary Encoder

**Investigación sobre el uso y funcionamiento de los componentes:**

**DF Player Mini HW-247A:** El DFPlayer Mini es un módulo compacto y versátil para reproducción de audio que soporta formatos comunes como MP3, WAV y WMA. Está diseñado para ser fácilmente integrable en proyectos de electrónica con o sin un microcontrolador gracias a su capacidad de trabajar con Arduino u otros sistemas de desarrollo. Además, cuenta con una ranura para tarjeta SD, un amplificador integrado para conectar directamente a un altavoz, y múltiples opciones de control incluyendo botones en el propio módulo para operación manual.



link de referencia: <https://www.tecneu.com/blogs/tutoriales-de-electronica/como-utilizar-el-dfplayer-mini-con-arduino-para-proyectos-de-audio?srsltid=AfmBOoqDybaz4q-ACsPKNmz3g6xhpHmT4682DxdIV2W_-5HpBiV0k_t->

**Sensor Touch:** Es un sensor de toque capacitivo diseñado para detectar interacciones táctiles sin la necesidad de contacto físico directo. Este sensor de un solo canal proporciona una interfaz intuitiva y fácil de implementar en una variedad de proyectos electrónicos.



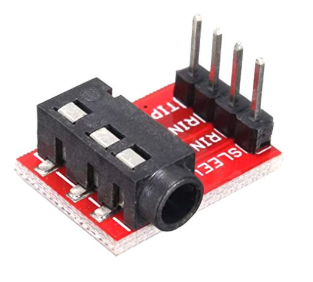
link de referencia:

<https://transtronix.com.mx/productos/modulo-touch-capacitivo-ttp223/?variant=805145516&pf=mc#:~:text=Descripci%C3%B3n%3A%20El%20TTP223%20es%20un,una%20variedad%20de%20proyectos%20electr%C3%B3nicos>.

no ayudó mucho pero adjunto el link:

<https://techmake.com/blogs/tutoriales/empezando-con-arduino-2d-sensor-touch>

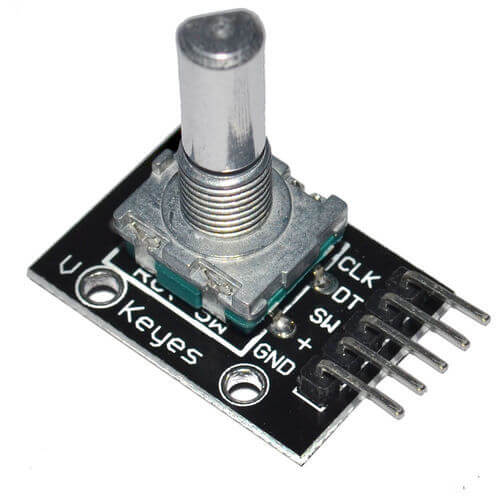
**TRRS Audio Jack:** Los conectores TRRS son los conectores de estilo de audio que se ven en algunos teléfonos, reproductores de MP3 y placas de desarrollo. TRRS significa "Tip, Ring, Ring, Sleeve" (punta, anillo,anillo, manga), lo que refleja el hecho de que, a diferencia de un conector estéreo estándar, éste tiene realmente tres conductores y una tierra.



link de referencia:

<https://solectroshop.com/es/modulos-pcb/1666-adaptador-audio-jack-hembra-35mm-trrs-con-placa-pcb.html>

**Rotary Encoder:** Es un dispositivo electromecánico que convierte el movimiento de giro de un eje en una señal *(Que puede ser analógica o digital dependiendo del tipo)* y genera una señal proporcional al giro de dicho eje que es muy útil para variar entre escalas musicales

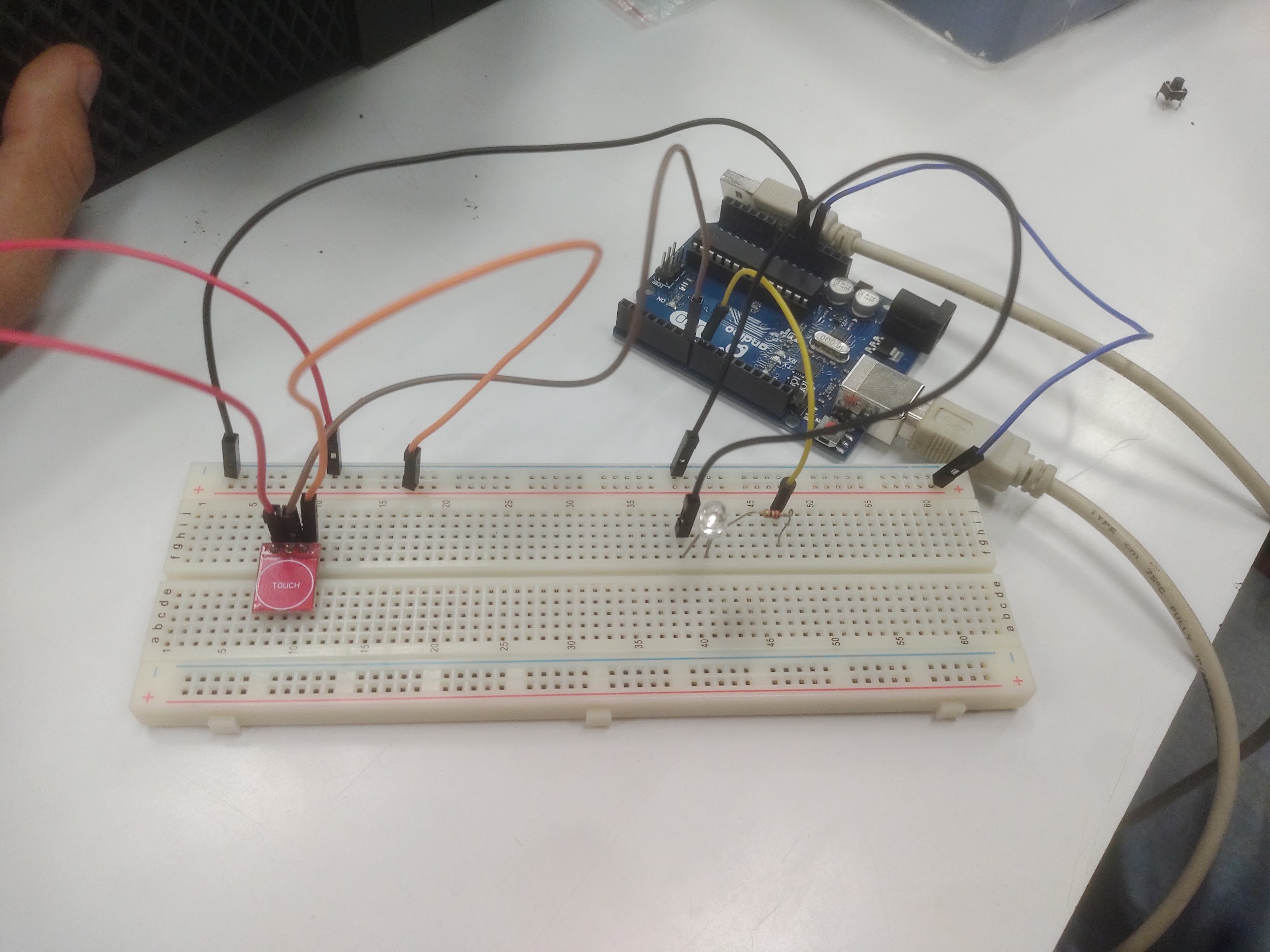


link de referencia:

<https://www.prometec.net/rotary-encoders/>

**Acá empezamos a experimentar sobre el uso del Sensor Touch:**

| #define led 8 #define sensorT 7  void setup() { pinMode (led, OUTPUT); pinMode (sensorT, INPUT); } void loop() { if (sensorT == HIGH) { digitalWrite(led, HIGH); } else{ digitalWrite(led, LOW); } } |
| --- |



Imagenes del compañero Jonas implementando el uso de los touch en el proyecto con Mateo



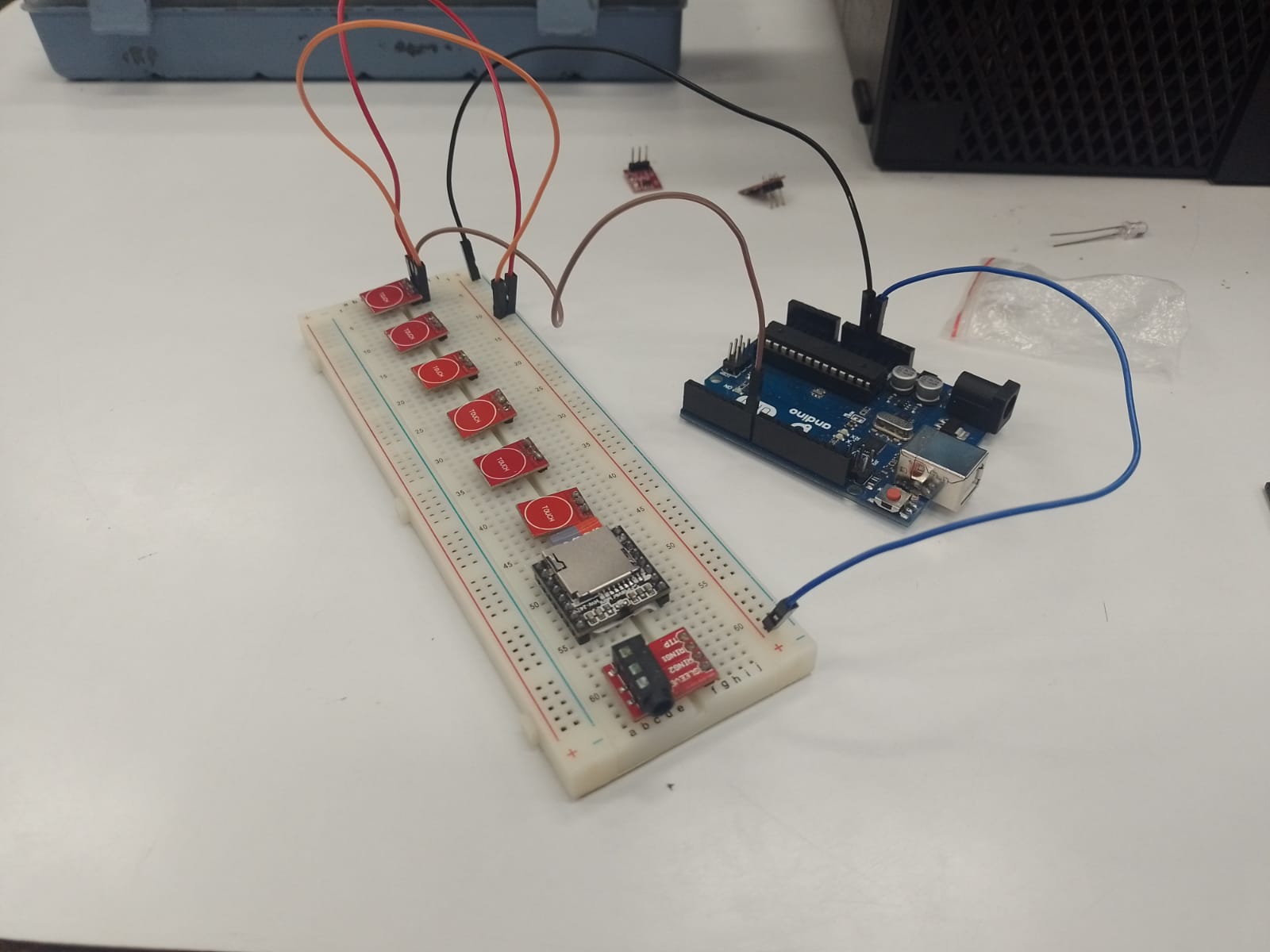
Nos dimos cuenta (nos dijo el profe) que nos faltaba leer el sensor entonces lo corregimos de esta manera poniendo “digitalRead(sensorT)”

Código corregido:

| #define led 8 #define sensorT 7  void setup() {  pinMode(led,OUTPUT);  pinMode(sensorT,INPUT); }  void loop() {  if (digitalRead(sensorT) == HIGH){  digitalWrite(led, HIGH);  }  else{  digitalWrite(led, LOW);    }  delay(100);  } |
| --- |

El profesor nos explicó un poco el funcionamiento de la MicroSD del DF Player Mini HW-247A y cómo implementarlo en el proyecto, nos queda investigar como es el uso del mismo(lo puse más arriba en la parte de investigación sobre el uso y funcionamiento de los componentes ).

En la siguiente imagen procedimos a conectar los touch para que más adelante hacer que cada vez que se toque algún touch(tecla) suene el sonido correspondiente a la misma desde el DF Player Mini



Investigue cómo exportar los tonos para más adelante subirlos al MicroSD con una Notebook:

Como subir musica a un microsd en pc:

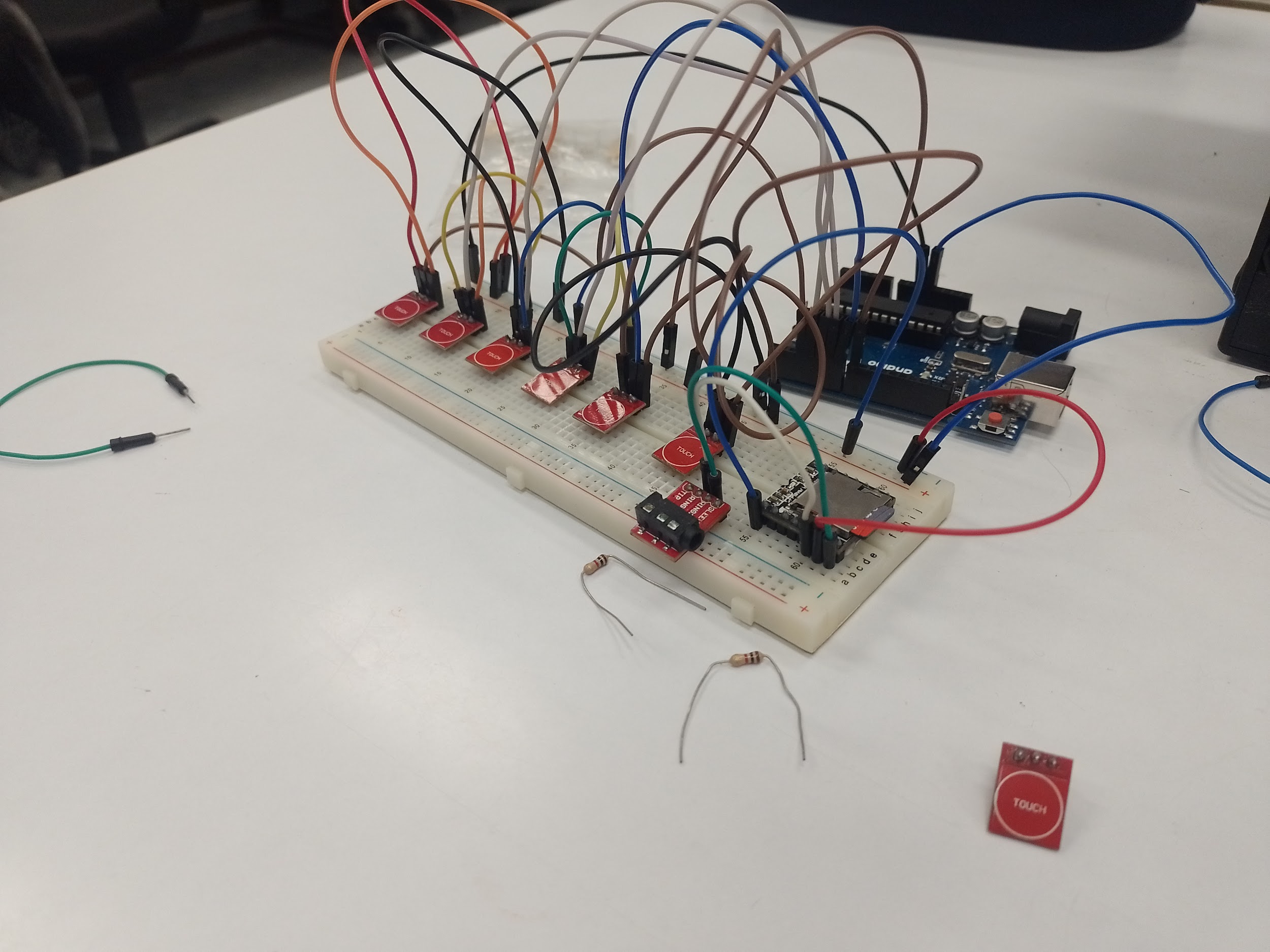
link

:<https://www.youtube.com/watch?v=0BCpmH0ljeU>

Con el video procedimos ya visto procedí con mi compañero Jonas Pereyra a descargar e instalar los archivos en MP3 de las notas principales del piano(do, re, mi, fa, sol, la, si), aunque tuvimos un problema y era que la PC no detectaba el MicroSD que lo intentamos solucionar con un video de YouTube pero no pudimos entonces llamamos al profe para que nos solucione el error.

Una vez ya solucionado el error guardamos las notas en el MicroSD con nombres como “001”, “002” y así sucesivamente

En la siguiente imagen ya están todos los sensores conectados y ya separamos las resistencias de un 1k Ohm para seguir la próxima clase



Ya dejamos hechas las conexiones y funcionando los touch, aunque falta todavía que se reproduzcan los sonidos de las notas dependientes a cada tecla

Para pasar audios mp3 de piano al DFplayer: <https://wiki.dfrobot.com/DFPlayer_Mini_SKU_DFR0299>

<https://www.tecneu.com/blogs/tutoriales-de-electronica/como-utilizar-el-dfplayer-mini-con-arduino-para-proyectos-de-audio>

22/11

Hoy apenas llegamos 2 miembros del equipo se encargaron de testear esta nueva funcionalidad y seguir con el código mientras yo me encargaba de anotar todo y lo hecho y re armar las conexiones en la protoboard y el arduino ya que nos lo habían desarmado totalmente.

En esta clase el profesor además de ya de por sí darnos componentes nuevos desde el principio hoy nos dio un nuevo componente que debemos implementar al proyecto llamado “Rotary Encoder”.

Su funcionalidad en el proyecto va a ser darle distintos “modos” a los sonidos de las teclas. La cantidad de modos que van a tener son 3

Link de informacion de como implementar el Rotary Encoder:

<https://arduinogetstarted.com/tutorials/arduino-rotary-encoder>

Procedí a implementar el DF Player Mini HW-247A. Para hacerlo funcionar era necesario que descarguemos una librería llamada “DFRobotDFPlayerMini” en el programa arduino.

Link de información:

<https://www.tecneu.com/blogs/tutoriales-de-electronica/como-utilizar-el-dfplayer-mini-con-arduino-para-proyectos-de-audio>

**Configuración de Pines para Sensores Táctiles**

| #define TOUCH\_PIN\_2 3 #define TOUCH\_PIN\_4 5 #define TOUCH\_PIN\_3 4 #define TOUCH\_PIN\_5 6 #define TOUCH\_PIN\_6 7 #define TOUCH\_PIN\_7 8 |
| --- |

#define: Define constantes para los pines de entrada asociados a los 7 sensores táctiles.

NUM\_KEYS 7: Indica el número total de teclas (o sensores táctiles) conectados.

Cada sensor táctil está asociado a un pin digital específico (pines 2 al 8).

| int touchPins[NUM\_KEYS] = {TOUCH\_PIN\_1, TOUCH\_PIN\_2, TOUCH\_PIN\_3, TOUCH\_PIN\_4, TOUCH\_PIN\_5, TOUCH\_PIN\_6, TOUCH\_PIN\_7}; |
| --- |

Se declara un array touchPins que almacena los números de los pines asignados a los sensores táctiles, facilitando su manejo en el código.

**setup() - Configuración Inicial**

| void setup() {  Serial.begin(9600);  mySerial.begin(9600); |
| --- |

Serial.begin(9600): Inicia la comunicación serial para depuración con la computadora.

mySerial.begin(9600): Configura el puerto serial virtual para comunicarse con el DFPlayer Mini.

| if (!myDFPlayer.begin(mySerial)) {  Serial.println("Error al iniciar DFPlayer.");  while (true); } |
| --- |

myDFPlayer.begin(mySerial): Inicializa el módulo DFPlayer Mini.

Si la inicialización falla, se imprime un mensaje de error y el programa queda en un bucle infinito (while (true);).

| for (int i = 0; i < NUM\_KEYS; i++) {  pinMode(touchPins[i], INPUT); } |
| --- |

Configura todos los pines asociados a los sensores táctiles como entradas (INPUT).

**loop() - Lógica Principal**

| void loop() {  for (int i = 0; i < NUM\_KEYS; i++) {  if (digitalRead(touchPins[i]) == HIGH) {  myDFPlayer.play(i + 1);  delay(200);  }  } } |
| --- |

**1. Bucle para Sensores Táctiles:**

El código recorre el array touchPins, leyendo el estado de cada pin asociado a los sensores táctiles.

digitalRead(touchPins[i]): Devuelve el estado del pin (HIGH o LOW). Si un sensor táctil detecta un toque, el pin correspondiente estará en estado HIGH.

**2. Reproducción de Archivos de Audio:**

myDFPlayer.play(i + 1): Reproduce el archivo de audio correspondiente al número de sensor.

Ejemplo: Si se activa el sensor táctil conectado a TOUCH\_PIN\_1 (índice 0 en el array), el archivo reproducido será track001.mp3. El DFPlayer numera los archivos en su tarjeta SD comenzando desde 1.

**3. Pausa para Evitar Lecturas Repetidas:**

delay(200): Introduce una pausa de 200 ms para evitar que el mismo toque se detecte varias veces consecutivas.

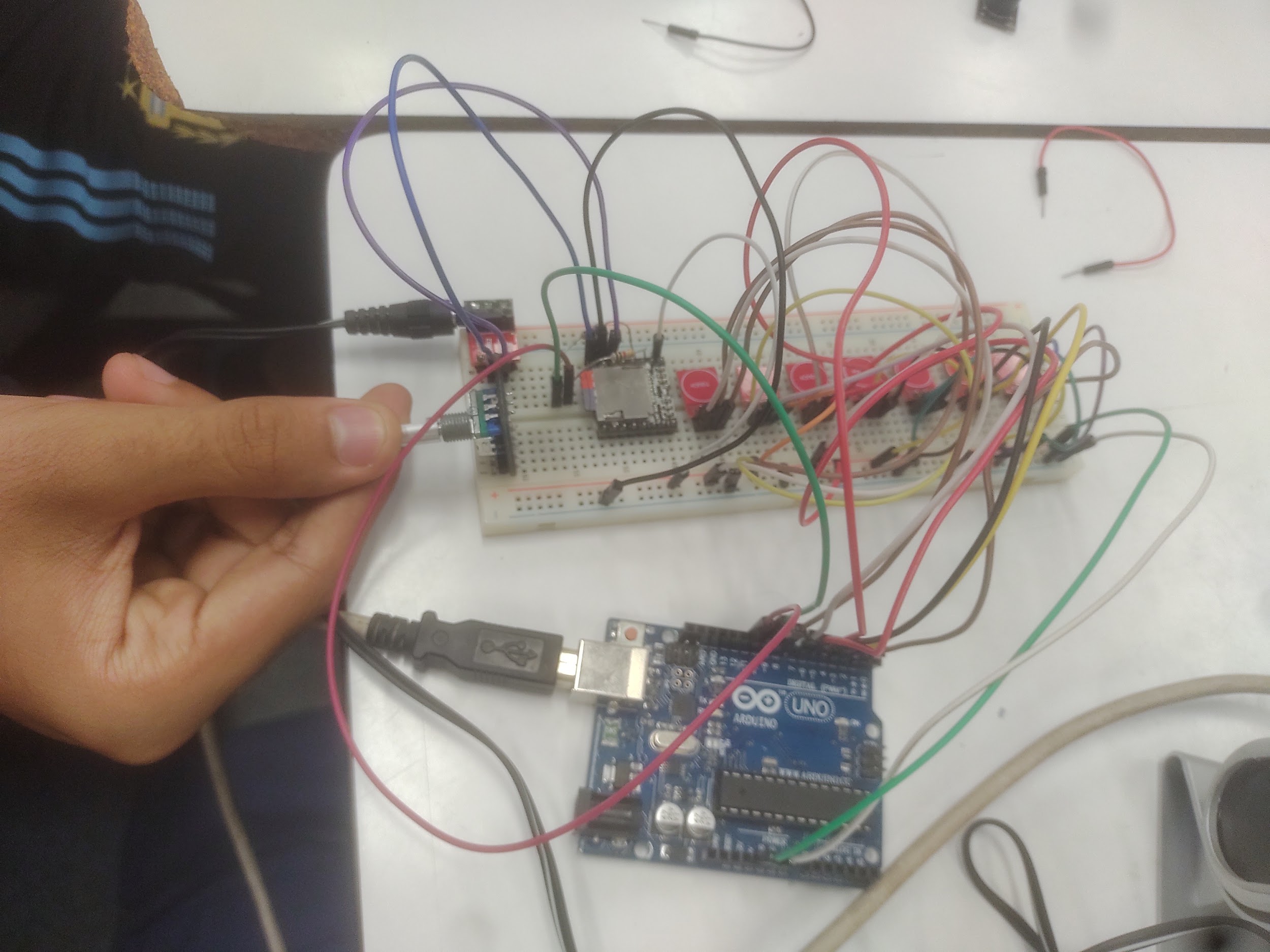
También intenté cambiar las notas del MicroSD ya que algunas se escuchaban muy graves y otras apenas agudas. Para esto fuimos al laboratorio con Metro y Juan pero cuando intente remover los archivos del MicroSD no nos dejaba, intente de todo hasta formatear el MicroSD para poder agregar de nuevo las notas pero era imposible ya que nos tiraba un error que decía que el MicroSD estaba bloqueado.Demasiado raro

Al final después de intentar de varios modos y ver algun video para solucionar el problema no pudimos y lo vamos a tener que dejar para la siguiente clase

Aca ya estábamos implementando el Rotary Encoder en el proyecto, aunque había un problema y era que al girarlo llegaba hasta 70 entonces lo que propuso Mateo para solucionarlo fue cambiar el máximo a 14.

Entonces asi quedaria **modo 1** en escala **0**, **modo 2** en escala **7** y el **modo 3** en escala **14(**ya que tiene que tener 3 modos).

Solo falta agregar más notas para los distintos modos ya que en total van a se 21 notas, 7 por cada modo



Hoy tuvimos bastantes problemas a comparación de otros días. Los problemas que tuvimos fueron:

1. Arduino no andaba, esto era asi ya que nuestro arduino no estaba en nuestra caja, entonces lo que hicimos fue agarrar uno que andaba por ahí, pero se nos hacia raro ya que nuestro proyecto no funcionaba y no habíamos cambiado nada, después de estar 20-30 minutos encontrando el error el profe nos dijo que podía ser el arduino ya que había uno por ahí que no andaba(era justo ese). Entonces buscamos otro arduino y pudimos solucionar ese “problema”
2. Las resistencias de RX y TX estaban tocándose, causando interferencia que impedía la reproducción de sonido.Se modificaron las conexiones de las resistencias y se separaron correctamente, eliminando la interferencia y permitiendo que el piano emitiera sonido.
3. El último sensor touch no andaba, esto fue debido a que ese tocuh nuestro del proyecto lo habían removido y lo reemplazaron por uno que no andaba(pensábamos que podía ser algún cable ya que anteriormente nos pasó que un touch andaba incorrectamente debido a un cable que funcionaba mal )

Luciano me contó sobre una idea que había tenido y entonces la planteamos y que si la podíamos llevar a cabo iba a ser mucho más fácil el implementar las demás notas, que consiste en que detecte los touch como notas de piano y así ir variando entre las escalas.

Luciano le preguntó a Consorti sobre la idea y dijo que era buena ya que los de la noche lo habían hecho de esa manera y aprobaron. Se lo planteamos al equipo pero por falta de tiempo decidimos no hacerlo ya que no sabíamos si lo íbamos a lograr

**Errores:**

* Los touch pueden que estén muy juntos ya que si queres presionar el borde de una, el touch de al lado tambien lo detecta
* No se pueden presionar 2 touch a la vez porque si no se rompe el sistema

**Soluciones:**

* Reorganizamos las conexiones y aumentamos el espacio entre los sensores para evitar inconvenientes
* Se cambió el código para permitir que se puedan presionar múltiples touch a la vez sin necesidad que se trabe el sistema

Y hasta ahora todo funciona como debería, no creo que nos veamos en la necesidad de cambiar grandes cosas a este punto, solo pulir si hay ciertos detalles nomas, y el proyecto ya está casi finalizado.

El equipo fue muy bueno ya que todos cooperaron para hacer de este un trabajo más sencillo y no tan difícil a pesar de que el profe nos replanteara de nuestra idea inicial(en ese momento Mateo se estresaba bastante), fue un trabajo bastante bueno para hacer en equipo y así va a concluir el proyecto si es que no hay más nada que agregar.