Universidad Francisco de Vitoria

Fundamentos de la programación con Arduino

Práctica Final

[Por Diego Viñals Lage y Gonzalo Ruipérez Ojea]

Contenido

Explicación detallada de INPUT_PULLUP

INPUT_PULLUP es usado en esta práctica para el Botón insertado en el joystick, la función que tiene es decirle al sistema cuando pulsamos dicho botón y en el caso de tener programado alguna función, ejecutarla. Cuando el botón no esta pulsado la corriente está pasando (5v en este caso) y se representa en forma de 1. Cuando pulsamos el botón cortamos la línea de corriente y se nos devuelve un 0.

En la Práctica, hemos usado el botón para reiniciar la placa de Arduino, mediante código hemos programado que mientras el Botón nos devuelva 1 (no está pulsado), no ocurra nada y siga haciendo el resto de funciones. Cuando se nos devuelve 0 (está pulsado) le decimos a la placa que debe reiniciarse.

Por tanto INPUT_PULLUP es una función que le dice a Arduino que solo hay dos posibles modos: 1 o 0 (HIGH or LOW) y nos da la posibilidad de decirle qué hacer en HIGH y que hacer en LOW

Explicación detallada de la función bitSet

La función bitSet la hemos usado en esta práctica para poder encender las luces. Como hemos usado un Registro de desplazamiento 74ch595, no podíamos acceder a cada bombilla por separado, pero con la función bitset() podíamos activar cada uno de los bits, y luego pasarlo al registro de desplazamiento, para así encender las bombillas que quisiéramos.

Si el número en la función era el 2, quiere decir que en el byte se activa el bit número 2. Así, con un loop for íbamos encendiendo cada uno. El problema que nos daba era que una vez lo encendias, se quedaba encendido, ya que no cambiada el bit a 0. La solución que encontramos fue usando una función asociada a bitset(), bitClear(). Esta función recibe los mismos parámetros que la función biSet(), pero hace lo contrario, en vez de encender el bit solicitado, lo que hace es apagarlo. De esta forma, encendiamos uno con bitset() y luego apagamos los anteriores con bitClear().

Esquema de conexiones del joystick analógico

El joystick tiene 5 pines. Uno de esos pines es el VCC, este pin lo tenemos que conectar a la placa de Arduino a 5V. Esto es lo que le da el voltaje al joystick para que pueda funcionar. El pin GND es el pin que se conecta a tierra en la placa arduino para poder cerrar el circuito. Los otros tres restantes son para leer los valores del eje x, del eje y, y el botón de pulsación.

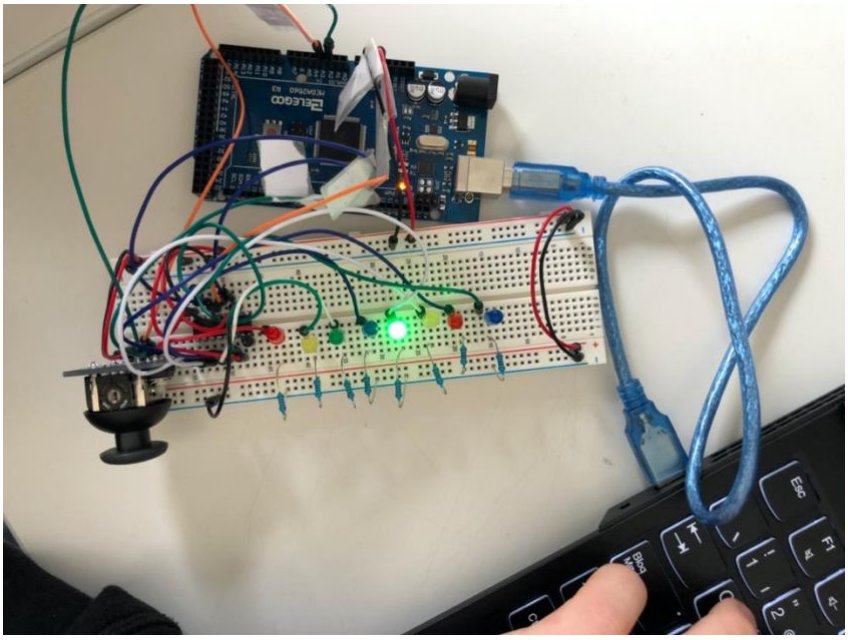
El pin VER/Y es el pin que conectamos a un pin analogico en la placa arduino, y que mediante la función analogRead() leemos los valores. Estos valores van entre 0 y 1024. Y ya con esos valores hacemos que las luces vayan de derecha a izquierda o de izquierda a derecha según corresponda.

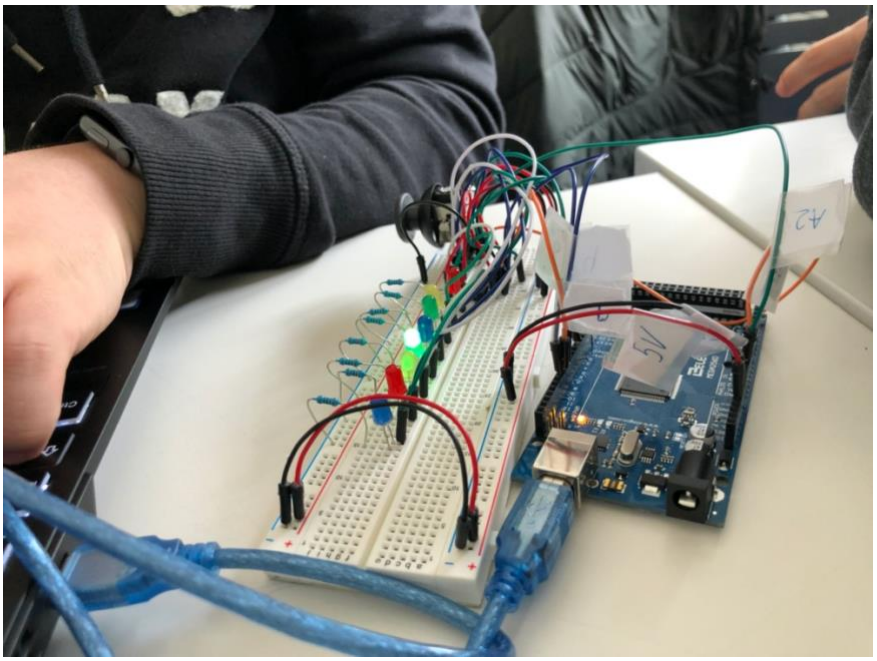
El siguiente pin es el VER/X, este pin lo conectamos a un pin anologico en la placa arduino, y como el pin de VER/Y, mediante la función analogRead() leemos los valores de x.

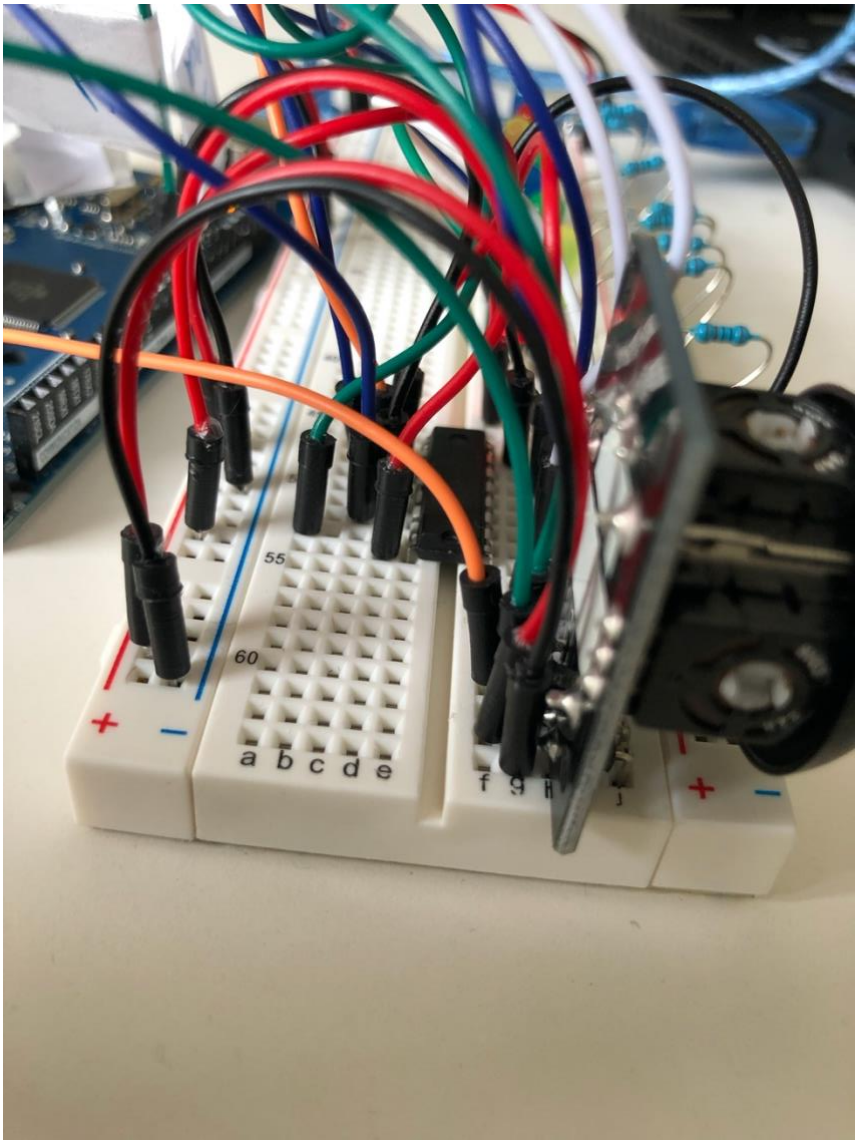
El pin que nos falta es uno que se conecta a un pin anolgico y nos dice si el boton está siendo pulsado o no. Mediante la función digitalRead() leemos el valor, y si el valor es 1 es que el botón está siendo pulsado, si el valor es 0, es que no está siendo pulsado.

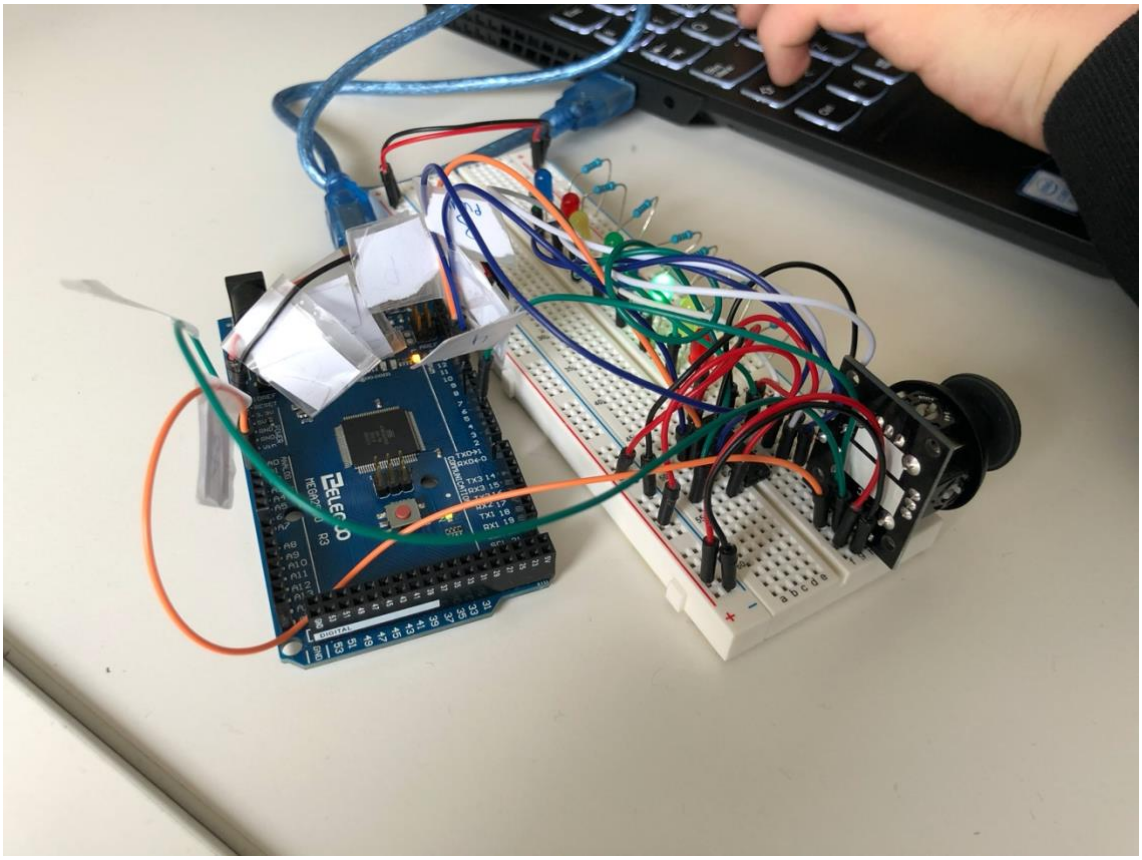
Diseño del circuito

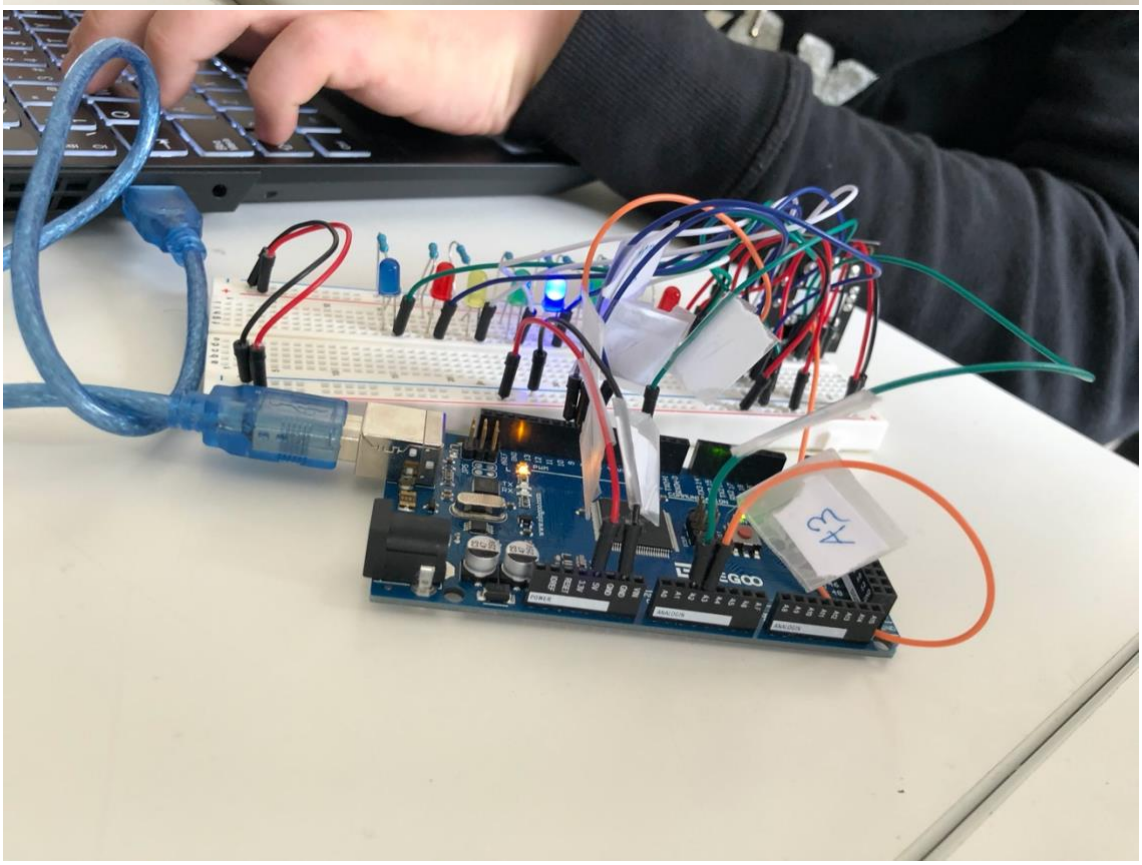
Hemos decidido hacerlo en físico, para poder verlo mejor y así poder usar el joystick y no dos potenciómetros como se haría si fuera en tinkercad.

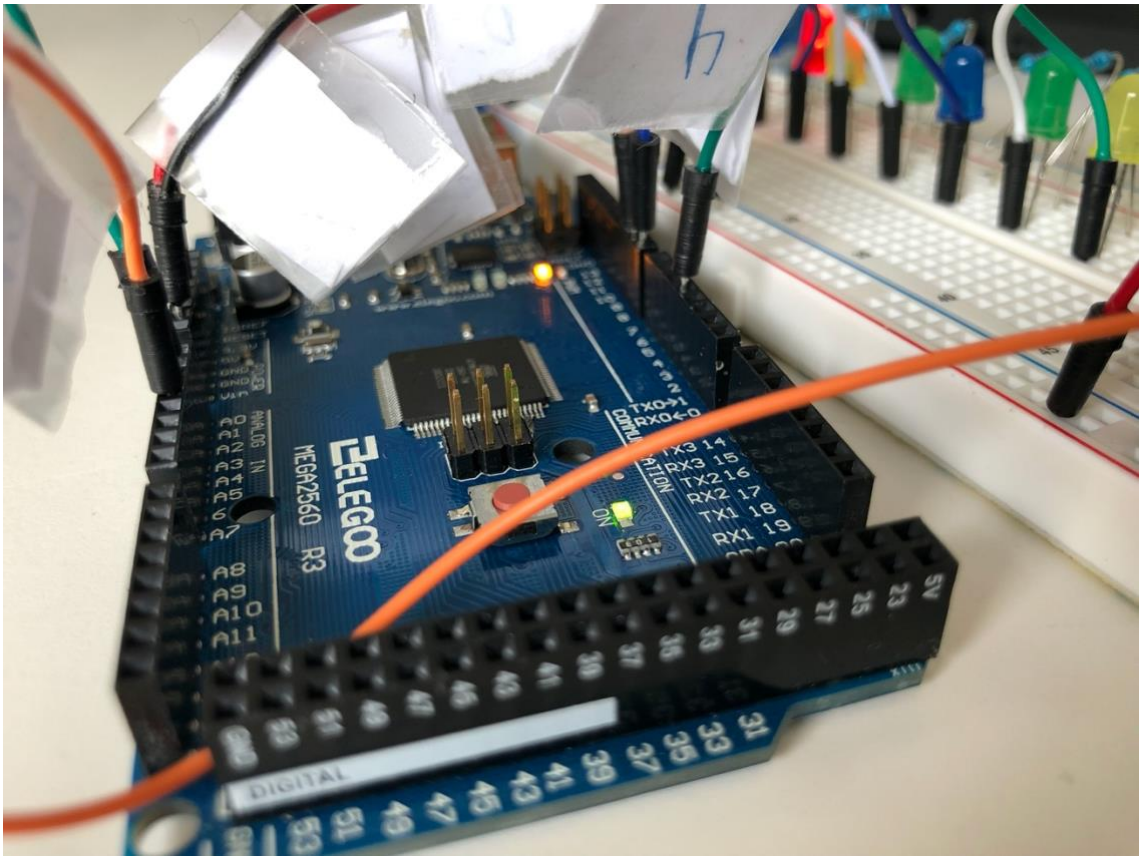





55
60
a b c d e
f g h i j
+
−
+
−






ELEGOO
MEGA2560 R3
DIGITAL
ANALOG IN
ON