MEMORIA PROYECTO SMP

Irene Huertas González

DNI: 15429589Q

Correo: irenehg@correo.ugr.es / e.irenehg@go.ugr.es

➔ Descripción funcional del sistema

Mi proyecto es un marcador que cuenta tiros libres acertados en una canasta de baloncesto. Con ayuda de sensores detecta cuando se ha encestado la pelota y va incrementando el valor del contador. Tiene un botón de reseteo para poner el contador a cero y guarda el valor que tenía el contador cuando lo apagas para que puedas seguir por donde lo dejaste en caso de quedarse sin batería o tener que interrumpir la actividad por cualquier otro motivo. Al encestar se activa una luz y un buzzer que ejecuta una melodía característica y distinta a la que suena cuando se resetea el marcador.

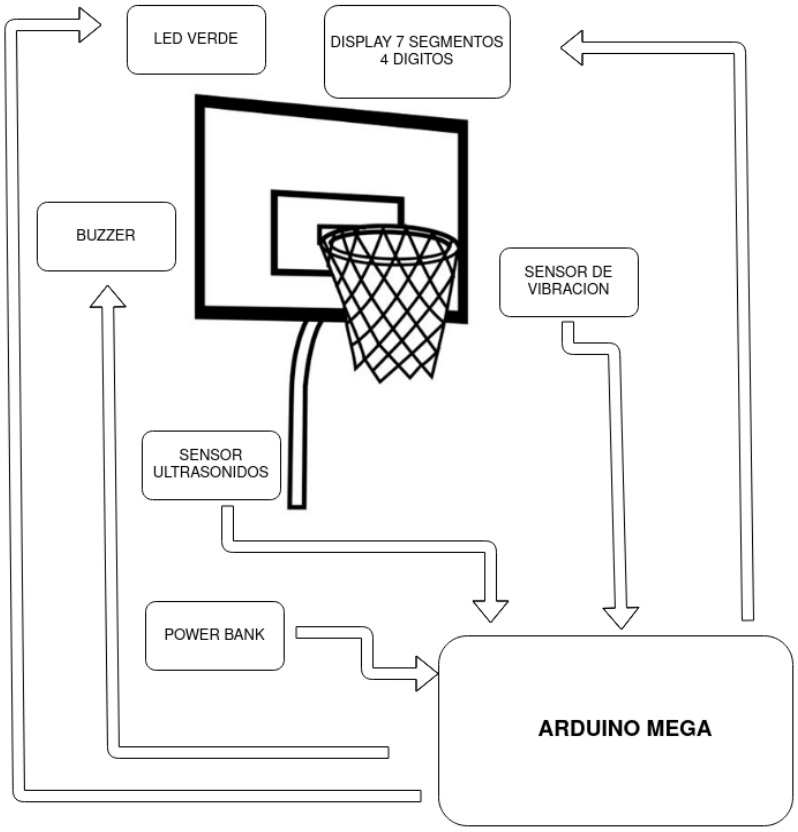
Mi proyecto está inspirado en este que aparece en el siguiente enlace:

Smart Basketball Scoreboard - Arduino Project Hub

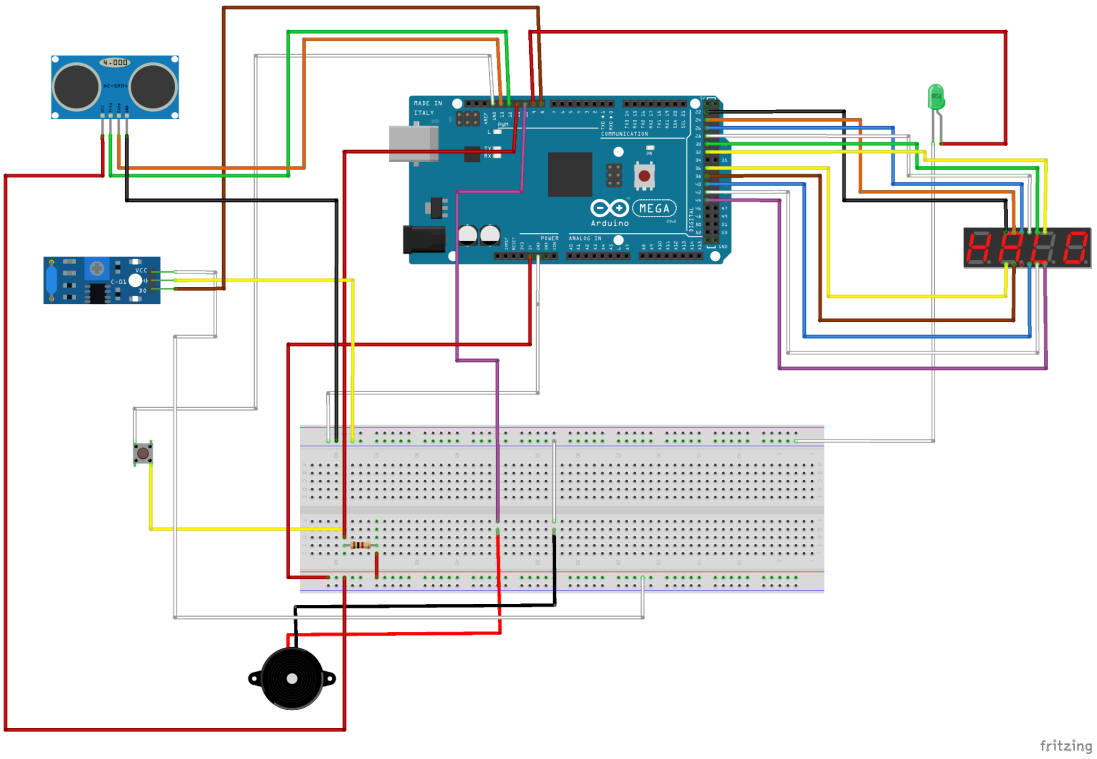
Además del código, presentan las siguientes diferencias:

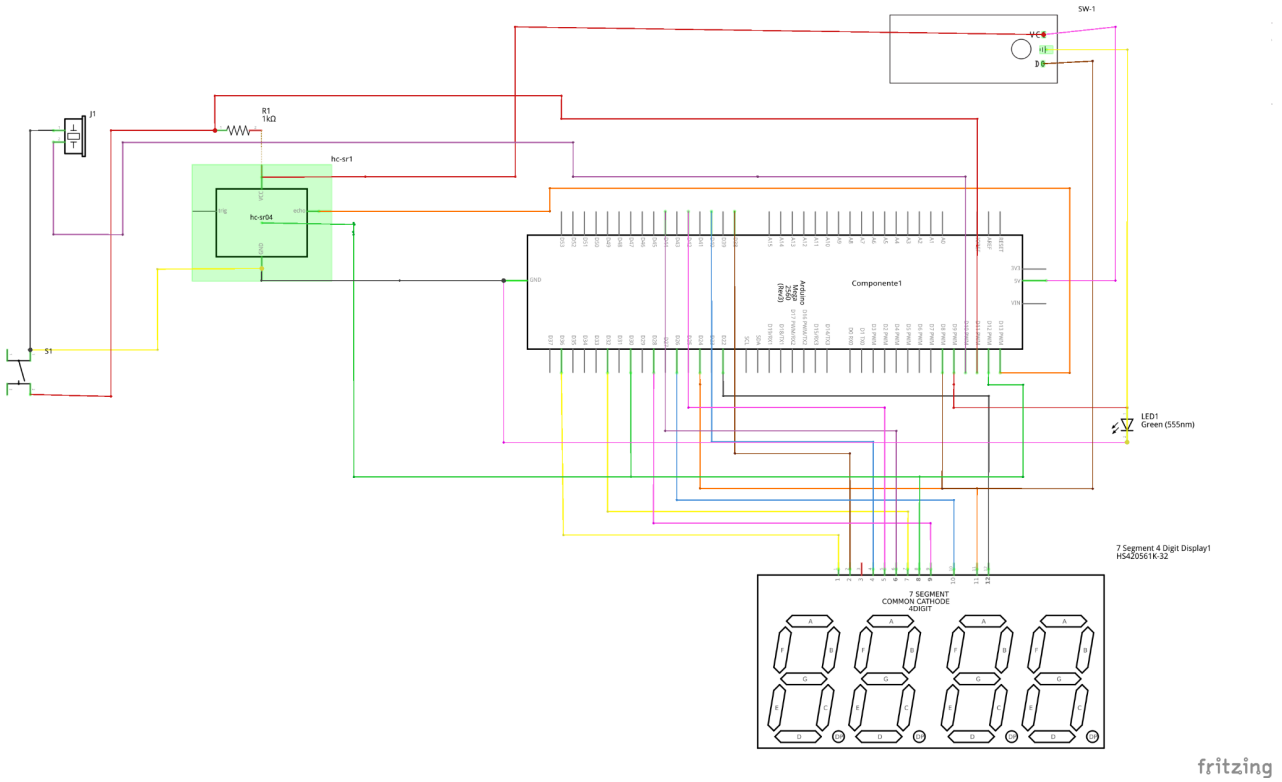
Diferencias	Mi Proyecto	Arduino ProjectHub
Sensor de proximidad	HC-SR04	E18-D80NK
Marcador	Display 4 dígitos	Móvil con aplicación (Bluetooth)
LED	LED verde	8mm RGB LED
Extras	buzzer	sensor de humedad y temperatura
EEPROM interna	Sí	No
Incrementar marcador de forma manual	No	Sí, tanto para canastas acertadas como fallidas
Canastas fallidas	No las cuenta	Las cuenta
Alimentación	Power Bank	Pila 9V

➔ Diagrama de bloques



➔ Representación visual del circuito y esquemático





➔ Lista de componentes

Kit de iniciación Arduino MEGA	25,30€
Cables	
Resistencia de 1KΩ	
Sensor de ultrasonidos HCSR04	
Display con 4 dígitos de 7 segmentos	
Led verde	
Pulsador con embellecedor	
Protoboard	
Buzzer	
Sensor de vibración SW-410 NC	4,99€
Paquete extra de 40 cables macho-hembra	3,99€
Tablero de baloncesto de pared	9,99€
Hilo de pescar	2€
Total del presupuesto empleado:	46,27€

➔ Recursos internos

Empleo la memoria EEPROM de Arduino para almacenar el contador que lleva la cuenta de tiros a canasta acertados y así, en caso de que se quede sin batería la fuente de alimentación o de tener que parar el juego, al reanudarlo el marcador continúe por el valor que iba.

También empleo un temporizador, es decir, mido el tiempo que transcurre desde que se detecta una vibración hasta que el sensor de movimiento detecta algo para que, en caso de producirse ambos en menos de 3 segundos (es el tiempo que he considerado que tarda como mucho una pelota en tocar el tablero y pasar por el aro) se considere canasta. Sin embargo si detectara vibración pero sin movimiento o el movimiento lo captara después de 3 segundos lo considerará de tiros diferentes y en ambos casos sería canasta fallida.

➔ Descripción sencilla del programa

```
setup()
      inicializa pines output e input según el dispositivo al que
referencian
      contador = valor que haya en la dirección 0 de la EEPROM
```

```
pantalla()
      contador se divide en unidades, decenas, centenas, u.millar
      muestra7seg(unidades); //función usada en prácticas Tinkercad
      se activa el cuarto dígito del display
     muestra7seg(decenas);
      se activa el tercer dígito del display
     muestra7seg(centenas);
     se activa el segundo dígito del display
     muestra7seg(unidades de millar);
     se activa el primer dígito del display
```

```
has_encestado()
      si encesta entonces:
            contador ++
            encendemos LED
            cancion() //usa tone
      reseteamos valores y tiempos de proximidad y vibracion
```

```
check_proximidad()
      Si distancia < 15 cm
            proximidad = true
            si tiempo_proximidad es 0
                  tiempo_proximidad = micros()
```

```
check_vibracion()
      Si capta vibracion
            vibracion = true
            tiempo_vibracion=micros()
/* en vibración guardamos el tiempo independientemente de si ya tenía
un valor almacenado o no porque la vibración tarda en estabilizarse y
se pondrá a true mucho antes de que hayamos siquiera recogido el balón
y eso puede dar lugar a no contar futuras canastas */
```

```
loop()
      if b1 presionado:
            cancion_reset() // utilizando tone
      check_vibracion()
      check_proximidad()
      si vibracion y proximidad entonces:
            si diferencias de tiempo < 3 segundos
                  has_encestado(1)
            si no
                  has_encestado(0)
      pantalla()
      Actualizamos dir 0 EPPROM = contador //solo si ha cambiado
```

➔ Enlace a video demostrativo

https://drive.google.com/file/d/1bQJ_jLN-SHdUdcfwXwrFdo0v-838I0dF/view?usp=sharing