

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Periféricos y Dispositivos de Interfaz Humana

Grado en Ingeniería Informática

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Practica 2 : Experimentación con un sistema de microcontrolador : Arduino . Cruce de Semáforos



Félix Ramírez García

José Rubén Céspedes Heredia

Se trata de implementar un cruce de semáforos controlado por Arduino, para ello utilizaremos en el primer semáforo los pines 3 (led rojo), 4 (led ámbar), 5 (led verde), en el segundo semáforo utilizaremos los pines 6 (led rojo), 7 (led ámbar) y 8 (led verde). La secuencia de funcionamiento debe ser:

1.rojo 1–verde 2 durante 3 segundos

2.rojo 1–ambar 2 durante 500 ms

3.verde 1–rojo 2 durante 3 segundos

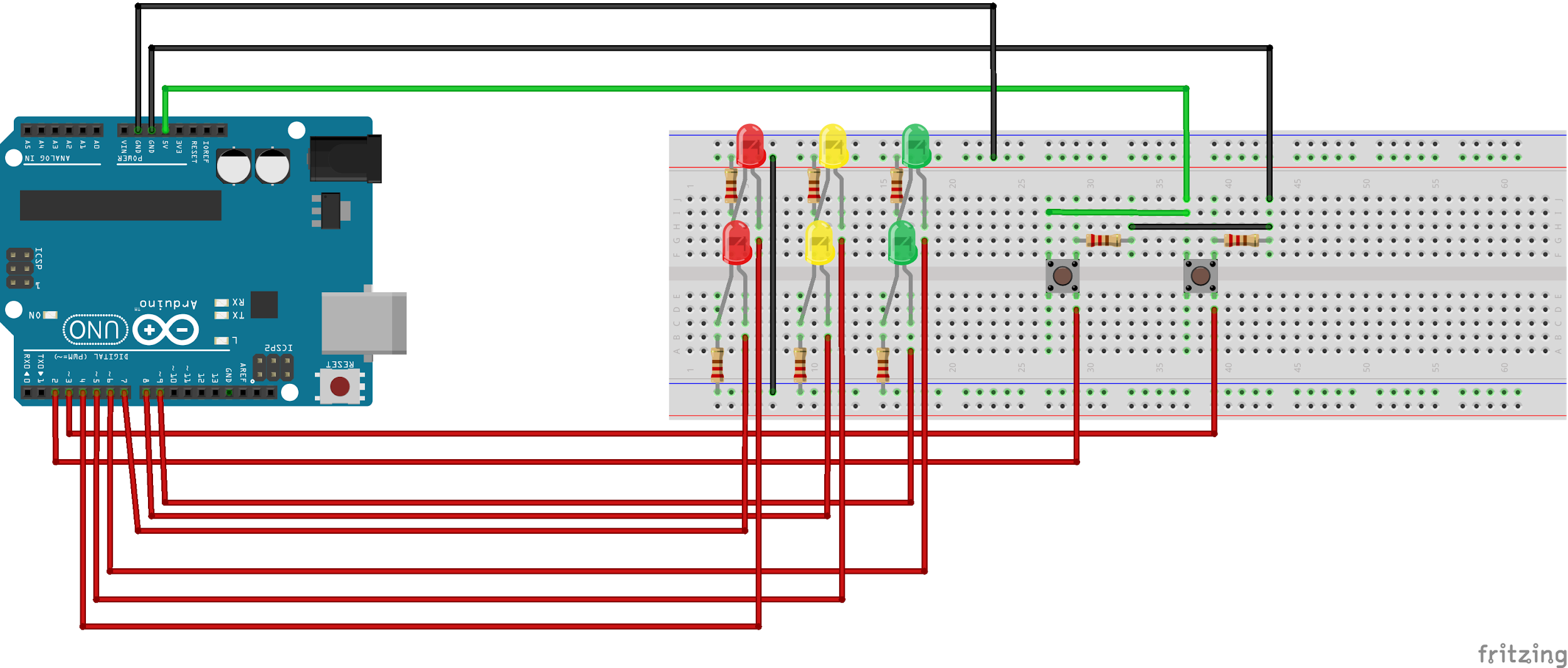
4.ambar 1–rojo 2 durante 500 ms

Adicionalmente, se agregaron dos botones para el control manual del semáforo:

Botón 1: conmuta entre modo automático temporizado o modo manual.

Botón 2: si el botón 1 se encuentra en modo manual, se produce un cambio del tráfico(el semáforo que estaba en verde pasa a ámbar [0.5s] y después a rojo , momento en que el que estaba en rojo pasa a verde).

**1.Imagen con el esquema de conexiones realizado en Fritzing.**

****

**2.Código fuente documentado. Describimos los pines de entrada y salida que se usan y su significado.**

**[code]  
/\* Definimos las variables CONSTANTES. Hemos reubicado los pines de los semaforos para poder usar los PINES 2 y 3 con INTERRUPCIONES \*/  
/\* PINES de los SEMAFOROS \*/  
const int pinRed1 = 4, pinAmber1 = 5, pinGreen1 = 6;  
const int pinRed2 = 7, pinAmber2 = 8, pinGreen2 = 9;  
  
/\* PINES de los BOTONES. El PIN 2 corresponde al boton que CONMUTA los MODOS MANUAL y AUTOMATICO. El PIN 3 es el que usa el peaton en modo manual para CAMBIAR EL SEMAFORO \*/  
const int button1Pin = 2;  
const int button2Pin = 3;  
  
/\* Definimos las variables NO CONSTANTES \*/  
volatile byte modeManual = LOW; // Indica el modo actual. LOW para automatico, HIGH para manual  
int transition = 0; // Si estamos en el modo manual, indica la transicion que se tiene que llevar a cabo cuando el peaton pulse el boton 2. 0 -> Nada; 1 -> de R1 a V1; 2 -> de R2 a V2  
boolean ejecutado = false; // Si estamos en el modo manual, indica si la transicion actual ya se ha ejecutado para que no se repita continuamente en el bucle  
  
void setup() {  
 // Todos los pines de los semaforos los ponemos como salidas  
 pinMode(pinRed1, OUTPUT);  
 pinMode(pinAmber1, OUTPUT);  
 pinMode(pinGreen1, OUTPUT);  
 pinMode(pinRed2, OUTPUT);  
 pinMode(pinAmber2, OUTPUT);  
 pinMode(pinGreen2, OUTPUT);  
  
 // Los pines de los botones los preparamos para la interrupcion  
 pinMode(button1Pin, INPUT\_PULLUP);  
 pinMode(button2Pin, INPUT\_PULLUP);  
  
 // A la interrupcion de cada boton le asignamos una funcion para cambiar la variable que le corresponda cada vez que lo pulsemos  
 attachInterrupt(digitalPinToInterrupt(button1Pin), changeMode, RISING);  
 attachInterrupt(digitalPinToInterrupt(button2Pin), changeTransition, RISING);  
}  
  
void loop() {  
 if(!modeManual){  
 // Estado automatico  
 modo\_automatico();  
 } else {  
 // Estado manual  
 modo\_manual();  
 }  
}  
  
void changeMode(){  
 modeManual = !modeManual;  
}  
  
void changeTransition(){  
 // La transicion va de 0 a 1. Una vez en 1 va de 1 a 2 y de 2 a 1 continuamente. Ponemos ejecutado a false porque la nueva transicion se debera de ejecutar (cuando se ejecute la volveremos a cambiar)  
 switch(transition){  
 case 0:  
 transition = 1;  
 break;  
 case 1:  
 transition = 2;  
 ejecutado = false;  
 break;  
 case 2:  
 transition = 1;  
 ejecutado = false;  
 break;  
 }  
}  
  
void modo\_automatico(){  
 // En ocasiones pasaremos del estado manual al automatico asi que apagamos los LEDs que se quedaron encendidos en dicho modo.  
 // Los LEDs seran distintos dependiendo de la ultima transicion que hayamos ejecutado  
 if(transition == 1){  
 digitalWrite(pinGreen1, LOW);  
 digitalWrite(pinRed2, LOW);  
 secuencia\_4();  
 } else if(transition == 2){  
 digitalWrite(pinRed2, LOW);  
 digitalWrite(pinGreen1, LOW);  
 }  
  
 // Ejecutamos la secuencia propia del modo automatico  
 secuencia\_1();  
 secuencia\_2();  
 secuencia\_3();  
 secuencia\_4();  
}  
  
void modo\_manual(){  
  
 // Comprobamos la transicion que nos toca realizar y si ya se ha ejecutado  
 switch(transition){  
 case 0:  
 // Si estamos en la transicion 0 es que venimos del modo automatico. Colocamos los LEDs en una posicion inicial de R1 encendido y V2 encendido  
 secuencia\_1();  
 break;  
 case 1:  
 if(!ejecutado){  
 // Si estamos en la transicion 1 y no la hemos ejecutado aun la ejecutamos. En este caso pasamos de R1 encendido a V1 encendido  
 secuencia\_2();  
 secuencia\_3();  
 ejecutado = true; // Cambiamos el valor de ejecutado para que en la proxima ejecucion del bucle no se vuelva a repetir la transicion de los semaforos  
 }  
 break;  
 case 2:  
 // Igual que el anterior pero pasando de R2 a V2  
 if(!ejecutado){  
 secuencia\_4();  
 secuencia\_1();  
 ejecutado = true;  
 }  
 break;  
 }  
}  
  
void secuencia\_1(){  
 digitalWrite(pinAmber1, LOW);  
 digitalWrite(pinRed2, LOW);  
 digitalWrite(pinRed1, HIGH);  
 digitalWrite(pinGreen2, HIGH);  
 delay(3000);  
}  
  
void secuencia\_2(){  
 digitalWrite(pinGreen2, LOW);  
 digitalWrite(pinAmber2, HIGH);  
 delay(500);  
}  
  
void secuencia\_3(){  
 digitalWrite(pinRed1, LOW);  
 digitalWrite(pinAmber2, LOW);  
 digitalWrite(pinRed2, HIGH);  
 digitalWrite(pinGreen1, HIGH);  
 delay(3000);  
}  
  
void secuencia\_4(){  
 digitalWrite(pinGreen1, LOW);  
 digitalWrite(pinAmber1, HIGH);  
 delay(500);  
}**

**3.Fotografías y/o vídeos demostrando el funcionamiento real del proyecto.**

<https://drive.google.com/open?id=1l7V8euvl22BL8taBy6vS-l7lkVIxDS4n>