**UNIVERSIDAD TECNOLÓGICA NACIONAL**

**INSTITUTO NACIONAL SUPERIOR DEL PROFESORADO TÉCNICO**



Proyecto

**Reloj con numeración binaria**

Trabajo práctico de Técnicas Digitales *3-603*

Castillo, Nicolás Alberto  
Legajo Nº 150526

nicolas.castillo@alu.inspt.utn.edu.ar

Fecha de presentación  
17/11/2023

Docente de la cátedra  
Prof. Ing. Israel Pavelek

# **1. Introducción**

En este proyecto se construyó un reloj con numeración binaria, basado en la placa de desarrollo Arduino UNO. El mismo consta de una serie de luces LED, conectados a registros de desplazamiento, que mediante su encendido y apagado expresan la hora en el sistema de numeración binaria.

El reloj empieza a contar desde la hora, minuto y segundo cero. Pero cuenta con dos pulsadores con los que se puede configurar la hora: con uno se selecciona el campo a modificar, y con el otro se suma un uno al campo correspondiente.

La finalidad de este proyecto fue combinar los conceptos de electrónica aprendidos en la cursada con los de informática, en primera instancia para fijarlos, pero también para mostrar cómo la simplicidad del sistema de numeración binario se puede traducir en una representación visual funcional de la hora.

El circuito funcionando se encuentra disponible en Tinkercad:

<https://www.tinkercad.com/things/kf2tOGETS7Z-reloj-con-numeracion-binaria?sharecode=pm1qLd24rhMyvz1zKC8jgqTnQUJaTAxAkxLh6ElukS8>

# **2. Componentes**

Los componentes necesarios para realizar este proyecto son:

* 1 placa Arduino UNO;
* 3 registros de desplazamiento 74HC595;
* 2 pulsadores;
* 17 LED: 5 rojos, 6 verdes y 6 azules; opcionalmente uno amarillo;
* 1 LED RGB;
* 19 resistencias: 2 de 10K ohms, 17 de 330 ohms; opcionalmente una más para el LED amarillo;
* 1 placa de desarrollo experimental o protoboard;
* 1 fuente de alimentación (preferentemente un dispositivo con salida USB);
* cables para conexión.

# **3. Diseño y funcionamiento**

Para mostrar la hora, sabiendo que tanto los segundos como los minutos están en el intervalo [0;59], alcanza con utilizar 6 LED para cada uno (60 < 26). En cambio, para contar las horas se necesita cubrir el intervalo [0;23], así que se necesitan 5 LED (24 < 25). Por lo tanto, para poder expresar completamente la hora, fue necesario utilizar 17 LED. Por una cuestión de claridad, se muestran los bits de la hora con color rojo, los minutos con color verde y los segundos con color azul.

Los LED fueron conectados a las salidas de los registros de desplazamiento. A su vez, los registros fueron conectados en cascada, es decir, la salida de datos serie del primero está conectado a la entrada de datos serie del segundo, y lo mismo entre el segundo y el tercero. Esto es para poder pasar en serie la información que se quiere mostrar, para que cada registro coloque en su salida la información en paralelo. Para lograr esto, fue necesario proveerle a los registros tres señales: un pulso de clock para pasar la información entre cada flipflop (SHI\_C), otro pulso de clock para poder pasar esa información a la salida de los registros (LAT\_C) y, en el caso del primer registro, la señal de información en serie (SER\_D). Estas señales fueron provistas por los terminales 10, 9 y 8 de la placa Arduino, respectivamente.

Para poder configurar la hora, un pulsador fue conectado al terminal 3, que es uno de los terminales disponibles para adjuntar una interrupción. Adicionalmente, se colocó un LED amarillo, un color distinto a los utilizados para expresar la hora, para poder comprobar que los pulsos lleguen correctamente a dicho terminal. El otro pulsador, que realiza la función de sumar uno al campo correspondiente, fue conectado al terminal 4 por una cuestión de cercanía, pero podría haber sido conectado a cualquier otro terminal disponible.

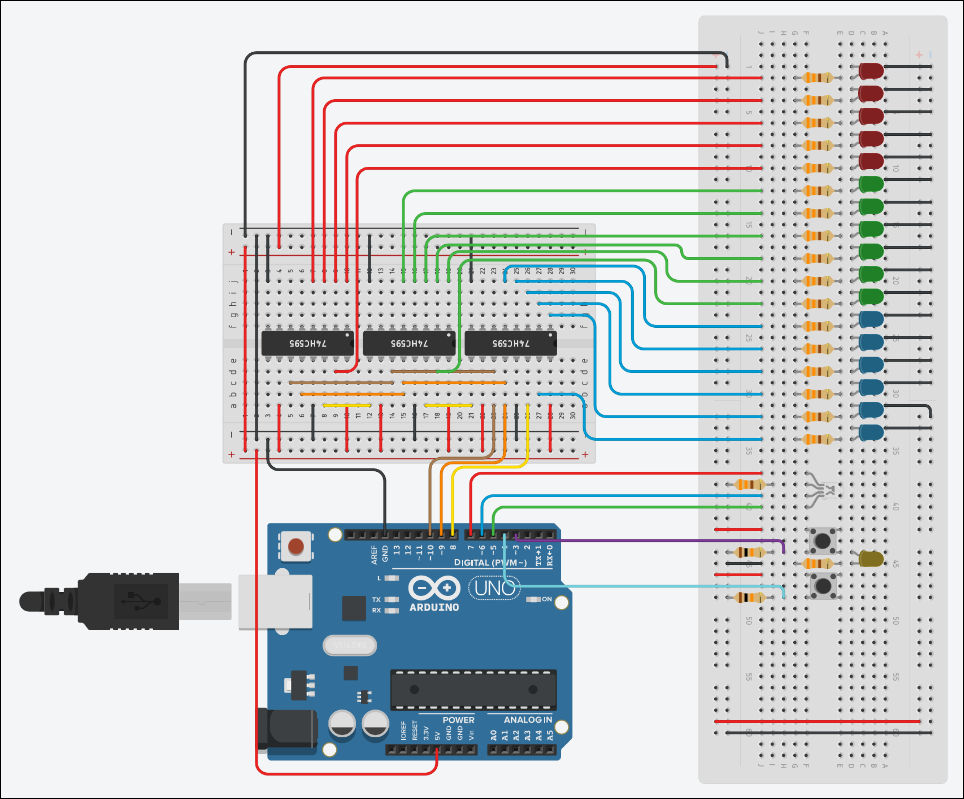
Para saber si el reloj se encuentra en modo configuración, se utilizó un LED RGB que fue conectado a los terminales 7, 6 y 5 de la placa Arduino. Este LED se ilumina del mismo color que el campo que está siendo configurado.

**4. Diagrama de bloques**

Escala de tiempo

Descripción generada automáticamente

# **5. Circuito**



# **6. Código**

// CONSTANTES

#define DR digitalRead

#define DW digitalWrite

#define MAX\_H 23

#define MAX\_M 59

#define MAX\_S 59

#define TMP\_CLK 1

#define TMP\_SEG 930

#define TAM 8

// TERMINALES

#define PUL\_1 3  // Interrupt para modo config

#define PUL\_2 4  // Pin para sumar 1 al campo

#define LED\_M 5  // Salida Verde led RGB

#define LED\_S 6  // Salida Azul led RGB

#define LED\_H 7  // Salida Rojo led RGB

#define SER\_D 8  // Serial Data

#define LAT\_C 9  // Latch Clock

#define SHI\_C 10 // Shift Clock

enum Campo {

    CS,

    CM,

    CH,

    MH

};

volatile uint8\_t estado;

volatile uint8\_t segs;

volatile uint8\_t mins;

volatile uint8\_t horas;

void pulso (uint8\_t pin) {

    milisegundos(TMP\_CLK);

    DW(pin, HIGH);

    milisegundos(TMP\_CLK);

    DW(pin, LOW);

}

void enviar (uint8\_t info) {

    for (int8\_t i = TAM - 1; i >= 0; i--) {

        DW(SER\_D, (info >> i) & 0x01);

        pulso(SHI\_C);

    }

}

void setup(){

    Serial.begin(9600);

    pinMode(PUL\_1, INPUT);

    pinMode(PUL\_2, INPUT);

    pinMode(LED\_H, OUTPUT);

    pinMode(LED\_M, OUTPUT);

    pinMode(LED\_S, OUTPUT);

    pinMode(SER\_D, OUTPUT);

    pinMode(LAT\_C, OUTPUT);

    pinMode(SHI\_C, OUTPUT);

    attachInterrupt(digitalPinToInterrupt(PUL\_1), config, RISING);

    DW(LED\_H, LOW);

    DW(LED\_M, LOW);

    DW(LED\_S, LOW);

    DW(LAT\_C, LOW);

    DW(SHI\_C, LOW);

    estado = Campo::MH;

    segs = 0;

    mins = 0;

    horas = 0;

}

void loop () {

    actualizarLEDS();

    imprimirHora();

    segs++;

    contar();

    milisegundos(TMP\_SEG);

}

void config () {

    estado = (estado + 1) % 4;

    milisegundos(100);

    if (estado != Campo :: MH) {

      actualizarLEDS();

      setupLedConf(estado);

          while (!DR(PUL\_1)) {

              if(DR(PUL\_2)) {

                  milisegundos(100);

                  confSum(estado);

              }

          }

          setupLedConf(3);

     }

}

void contar () {

    if (segs > MAX\_S) {

        segs = 0;

        mins++;

    }

    if (mins > MAX\_M) {

        mins = 0;

        horas++;

    }

    if (horas > MAX\_H) {

        horas = 0;

    }

}

void imprimirHora () {

    Serial.print(horas);

    Serial.print(":");

    Serial.print(mins);

    Serial.print(":");

    Serial.println(segs);

}

void actualizarLEDS () {

    enviar(horas);

    enviar(mins);

    enviar(segs);

    pulso(LAT\_C);

}

void confSum (uint8\_t i) {

    switch (i) {

      case 0:

        segs++;

        break;

      case 1:

        mins++;

        break;

      case 2:

        horas++;

        break;

    }

    contar();

    actualizarLEDS();

  }

void setupLedConf (uint8\_t i) {

    switch (i) {

      case 0: // Segundos

        DW(LED\_H, LOW);

        DW(LED\_M, LOW);

        DW(LED\_S, HIGH);

        break;

      case 1: // Minutos

        DW(LED\_H, LOW);

        DW(LED\_M, HIGH);

        DW(LED\_S, LOW);

        break;

      case 2: // Horas

        DW(LED\_H, HIGH);

        DW(LED\_M, LOW);

        DW(LED\_S, LOW);

        break;

      case 3: // Apagar

        DW(LED\_H, LOW);

        DW(LED\_M, LOW);

        DW(LED\_S, LOW);

        break;

      }

}

void milisegundos (uint16\_t n) {

    for (uint16\_t i = 0; i < n; i++) {

        delayMicroseconds(1000);

    }

}