

Informe de Análisis

Sistema de comunicación encriptado

John Edison Chamorro Coral

Daniela Alvarez Bernal

Departamento de Ingeniería Electrónica y
Telecomunicaciones
Universidad de Antioquia
Medellín
Febrero 21 de 2022

Índice

1. Resumen	2
2. Introducción	2
3. Objetivos Generales	2
4. Objetivos Específicos	2
5. Análisis del problema	3
5.1. ¿Cómo funciona el Integrado 54HC59?	4
5.2. ¿Qué es y cómo funciona Arduino?	8
5.3. ¿Cómo funcionan los puertos digitales de Arduino?	8
5.4. ¿Cómo generar una señal de reloj en Arduino?	9
5.5. ¿Cómo transmitir datos de manera serial por Arduino?	9
5.5.1. Inclusion de código de comunicacion serial simple entre dos Arduinos Uno R3	10
6. Marco Experimental	12
7. Resultado	12
8. Conclusiones	12

1. Resumen

El tratamiento de la información bancaria requiere de varias capas de seguridad aplicada desde todos los frentes. El canal de comunicación es uno de los puntos críticos respecto a la seguridad, pues es más vulnerable al estar expuesto sin seguridad constante. Es por ello que en este documento se procede a realizar una solución de seguridad, con la cual, por medio de microcontroladores Arduino Uno R3, se manipulará la información transmitida desde un punto y aplicando algoritmos de encriptación se brindará un nivel de protección y seguridad a dicha información hasta llegar al receptor.

2. Introducción

La información se ha vuelto uno de los objetos de mayor valor en la actualidad, pues con estos se puede realizar grandes tareas en pro del crecimiento investigativo, empresarial, entre otras áreas; Pero a su vez la vulnerabilidad de esta puede poner en conflictos a muchas personas e incluso a naciones. Es por ello que se ha incrementado la investigación de nuevas tecnologías que permitan proteger la información cuando sea requerido. Hay que tener en cuenta que la comunicación se puede transmitir por muchos medios [1], y por tanto tiene muchos frentes de ataque por lo que se hace necesario enfocarse en cada uno de ellos y brindar una solución a la problemática de la vulnerabilidad de la información. A pesar de que en la actualidad hay muchas herramientas que buscan la protección de datos, para el caso donde la comunicación se realizara a través de medio cableado, lo mejor para seguridad de la información es usar encriptación de datos. Es decir se modificará la información de tal forma que solo pueda ser entendida o cifrada por medio de un algoritmo especializado y único dentro del protocolo de comunicación que se estableció [2].

3. Objetivos Generales

Como objetivo general se requiere generar un canal de comunicación entre dos microcontroladores Arduino Uno R3, en donde uno de ellos se encargara de enviar información encriptada en serie y esta se paralizará para proceder a su desencriptación; El resultado de desencriptar será enviado en serie hacia el otro microcontrolador receptor quien desplegará en pantalla la información. La comunicación debe brindar un nivel de seguridad y fiabilidad de los datos, además de tener un comportamiento eficaz y ser un sistema escalable.

4. Objetivos Específicos

- Analizar y documentar el funcionamiento del integrado 54HC595 y cómo este puede ayudar en la transmisión de datos en paralelo.

- Analizar y documentar el funcionamiento de los puertos digitales del microcontrolador Arduino Uno R.
- Analizar y documentar como generar una señal de reloj por medio de Arduino Uno R3.
- Analizar y documentar cómo transmitir información serial por medio de un puerto digital de Arduino Uno R3.

5. Análisis del problema

Se requiere un sistema de comunicación síncrono, en donde se envíe una trama de datos de forma serial, y en el recorrido pase por un sistema el cual convertirá los datos en una trama paralela. Esa trama paralela será encriptada y enviada al sistema de desencriptación, el cual procederá a realizar la selección de los datos correctos y se procesará para serializarlos y enviarlos al receptor.

Bajo lo anterior el sistema se podría descomponer en los siguientes subsistemas:

- Un subsistema de transmisión de datos de manera serial en tramas de 8 bits.
- Un subsistema de Reloj para protocolizar la transmisión y recepción de datos.
- Un subsistema conversor de datos seriales a datos paralelos en tramas de 8 bits.
- Un subsistema de desencriptación de datos con entrada en paralelo y salida en serie.
- Un subsistema de recepción de datos de manera serial síncrono con el sistema de Reloj.

Las consideraciones a tener en cuenta para la solución de este problema, son:

- El ambiente de trabajo será virtual.
- Dentro de las herramientas de trabajo se tiene dos Arduino Uno R3 (transmisor y receptor), un dispositivo Integrado 54HC595, cables y placas.
- La trama de datos a enviar estará constituida por los datos a transmitir, y un dato bandera el cual indicará al sistema de desencriptación, que se debe realizar el proceso de toma de datos para su proceso de serialización.
- La información tratada será realizada bajo bloques de 8 bits.
- La serialización de datos y el sistema de reloj se realizará por medio de dos puertos digitales de Arduino Uno R3 transmisor.

- Para el procesamiento de los datos en paralelo se emplearan 8 puertos digitales del Arduino receptor.
- Lo descriptado se mostrara por consola como medio de representación visual.

5.1. ¿Cómo funciona el Integrado 54HC59?

Circuito integrado:

Un circuito integrado en una estructura pequeña de material semiconductor [9], normalmente de silicio, sobre la que se fabrican circuitos electrónicos. En cuanto a las funciones integradas los circuitos se clasifican en dos grandes grupos:

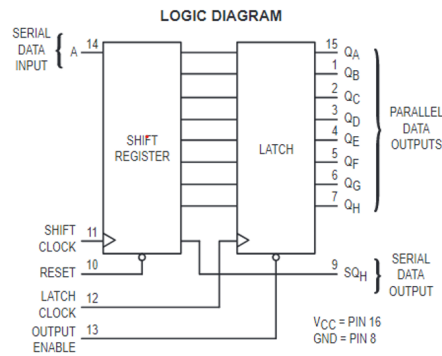
- Circuitos integrados analógico: No tratan con señales digitales sino analógicas, es decir que función abordando señales continuas variables en lugar de señales discretas, la electrónica digital es discreta por el hecho de que cada señal solo puede tener dos valores. En cambio, en la electrónica analógica cada señal es de rango variable.
- Circuitos integrados digitales o circuitos lógicos: Este tipo de circuito integrado tienen dos niveles definidos: 1 y 0, lo que implica que funcionan en matemáticas binarias en las que 1 significa encendido y 0 apagado. Estos circuitos emplean componentes encapsulados, los cuales pueden albergar puertas lógicas o circuitos lógicos más complejos. Los Circuitos Lógicos están compuestos por compuertas lógicas como la compuerta AND(Y), compuerta OR(O), compuerta NOT(NO) y otras combinaciones más complejas de los circuitos antes mencionados. Las compuertas lógicas tienen una única salida, aunque pueden tener una o más entradas. Los componentes que emplean estos circuitos están estandarizados, para que haya una compatibilidad entre fabricantes, de forma que las características más importantes sean comunes. De forma global los componentes lógicos se engloban dentro de una de las dos familias siguientes: TTL: diseñada para una alta velocidad. CMOS: diseñada para un bajo consumo, Esta familia requiere un manejo especial ya que la electricidad estática del cuerpo humano podría dañarlos al tocar sus terminales. Actualmente dentro de estas dos familias se han creado otras, que intentan conseguir lo mejor de ambas.

Cada circuito integrado tiene cierto número de pines o terminales. Es muy importante saber dónde va conectado cada terminal, ya que si se conecta en forma errada se puede dañar fácilmente. Los pines están numerados en el sentido contrario a las manecillas del reloj en forma de U. Los circuitos integrados vienen en configuraciones de 8, 14, 16, 18, 20, 24, 40 y 64 pines.

Circuito integrado 74HC595:

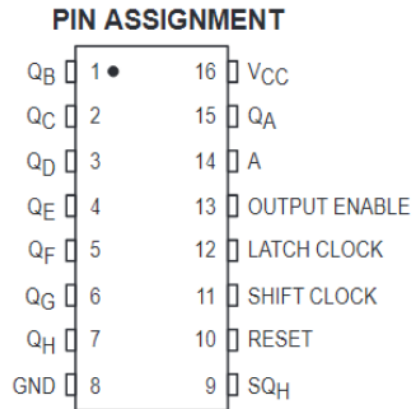
Es un registro de desplazamiento, es decir un circuito digital secuencial que cuenta con entrada en serie (bit a bit) y salida en paralelo de 8 bits. Esto quiere decir que las salidas en un instante dado no dependen única y exclusivamente de las entradas en dicho instante, sino que dependen también de las entradas que han ocurrido con anterioridad. En este tipo de registros con salida en paralelo se dispone de la salida de cada flip-flop por lo que una vez almacenados los datos cada bit se representa en su respectiva salida. De esta manera todos los bits de salida estarán disponibles al mismo tiempo.

Este circuito consistente en una serie de biestables o flip-flop (circuito que tiene dos estados estables y puede almacenar información.) del tipo D (biestable de datos o seguidor. Tienen una única entrada D), conectados en cascada y un latch (circuito biestable que puede almacenar un bit de información) tipo D (solo tiene una entrada D) de 8 bits. El registro de desplazamiento proporciona datos paralelos al latch y también cuenta con una salida en serie.



La principal diferencia entre el latch y el flip-flop es que un latch cambia de estado de inmediato, según sus señales de excitación de entrada, es decir que verifica la entrada continuamente y cambia la salida cuando hay un cambio en la entrada. [3] Mientras que un flip-flop espera la señal de su reloj antes de cambiar de estado. [4]

Descripción de los pines:



Inputs:

- A(Pin 14): Entrada de datos en serie. Los datos en este pin se desplazan al registro de desplazamiento en serie de 8 bits.

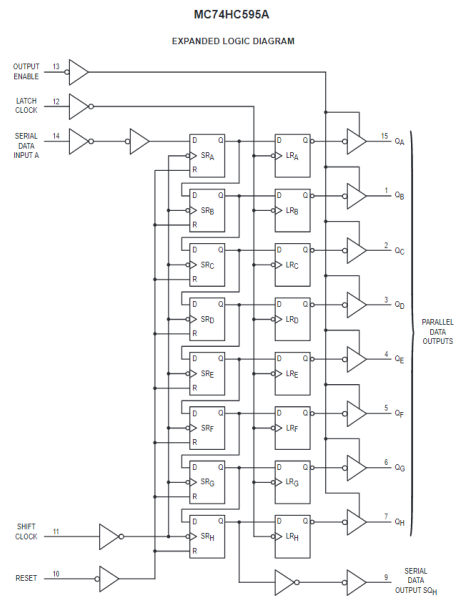
Control inputs:

- Shift Clock(Pin 11): Entrada de reloj de registro de desplazamiento. Una transición de bajo a alto en esta entrada hace que los datos en el pin de entrada en serie se desplacen al registro de desplazamiento de 8 bits.
- Reset (Pin 10): Alto-bajo asincrónico, entrada de restablecimiento de registro de desplazamiento. Un nivel bajo en este pin restablece solo la parte del registro de desplazamiento de este dispositivo. El latch de 8 bits no se ve afectado.
- Latch Clock (Pin 12): Entrada de reloj del almacenamiento latch. Una transición de bajo a alto en esta entrada bloquea los datos del registro de desplazamiento. Funciona para cargar los datos del pin 14 de datos, al momento que se activa con el estado lógico "1"
- Output Enable (Pin 13): Habilitación de salida activa-baja. Un nivel bajo en esta entrada permite que los datos de los latches se presenten en las salidas. Un valor alto en esta entrada obliga a las salidas (Q_A-Q_H) a entrar en el estado de alta impedancia (Resistencia de un circuito al flujo de una corriente eléctrica alterna). La salida serial no se ve afectada por esta unidad de control.

Outputs:

- Q_A – Q_H (Pins 15, 1, 2, 3, 4, 5, 6, 7): Salidas no invertidas (no son contrarias a su entrada) de 3 estados (un estado de bajo nivel (0), un estado de alto nivel (1), un estado de alta impedancia o estado flotante (Z)) de los latches.

- SQH (Pin 9): Salida de datos en serie no invertida. Esta es la salida de la octava etapa del registro de desplazamiento de 8 bits. Esta salida no tiene capacidad de tres estados. [5]



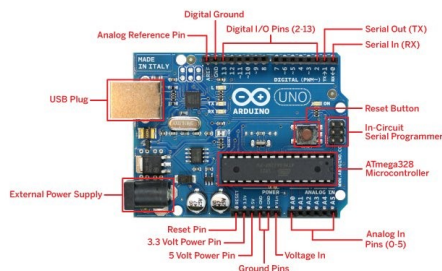
Especificaciones y características:

- Rango de voltaje de alimentación: $-0.5 V_{cc}$ a $+6 V_{cc}$.
- Voltaje de entrada: $0 V_{cc}$ a V_{cc} .
- Voltaje de salida: $0 V_{cc}$ a V_{cc} .
- Consumo de corriente: 80 A max.
- Frecuencia de reloj máxima opera dependiendo el VCC:
 - $2 V_{cc}$ la frecuencia es 4.8 ns.
 - $4.5 V_{cc}$ la frecuencia es 24 ns.
 - $6 V_{cc}$ la frecuencia es 28 ns.
- Familia lógica: CMOS.
- Número de bits: 8 bits.
- Número de entradas: 3.

5.2. ¿Qué es y cómo funciona Arduino?

Arduino es una placa electrónica de hardware libre, que cuenta con múltiples periféricos, los cuales interactúan como transmisores o receptores de información permitiendo manipular variables físicas o virtuales, a través de un microcontrolador que para este caso es ATmega328P [6]. Las variables se manipulan por medio del microcontrolador que a su vez es configurado por medio de instrucciones en lenguaje de programación de alto nivel [7].

Debido a su enorme flexibilidad y carácter libre y abierto, este dispositivo permite realizar casi cualquier cosa como por ejemplo relojes, básculas, robots, entre otras cosas.



5.3. ¿Cómo funcionan los puertos digitales de Arduino?

Los pines en Arduino son adaptados para trabajar como entrada o salida de datos, pero para que Arduino reconozca el puerto digital como entrada o salida debe ser configurado a través de código. Los pines configurados como entrada están en estado de alta impedancia [8], lo que quiere decir que se requiere muy poca corriente para poder realizar una lectura de la variable asociada a dicho puerto; el problema asociado a esa alta impedancia es que si ese puerto no tiene conectado nada, registra el ruido electrónico del entorno o se acopla capacitivamente al estado de un pin cercano [8]. Alguno de los puertos digitales de la placa permite realizar transmisión de PWM con una frecuencia de 490Hz y con una tensión de aproximadamente 2.5V [9]. Ahora bien el puerto digital configurado como salida tiene una baja impedancia lo que permite suministrar hasta 40 mA hacia otro circuito externo Tabla 1. Básicamente su funcionamiento es a través de corriente o tensión, interactuando con las variables o circuitos como si fueran switches. Es decir el pin se activa con un valor de tensión de 3V pero no se activa con un valor de tensión menor al valor antes mencionado; respecto a un puerto como salida, actúa con un nivel alto con una tensión de 5 V (0 3.3V para placas alimentadas a esta tensión), y con un valor bajo con 0V.

5.4. ¿Cómo generar una señal de reloj en Arduino?

Recordando que Arduino tiene en su estructura 6 puertos digitales Tabla 1, que me permiten realizar transmisión en PWM, es fácil deducir que se podría realizar una señal de reloj por medio de estos puertos. Bastaría con configurar el ancho de pulso requerido para el reloj; Teniendo en cuenta que el puerto permite enviar, en este formato PWM, valores entre 0 y 255 haciendo alusión al porcentaje que el valor alto de la señal permanecerá dentro del periodo de transmisión, es decir que si requiero un ancho de pulso del 50% Cabe resaltar que para cada placa de Arduino, la frecuencia de transmisión de datos por medio de estos puertos digitales varía; Para el caso de Arduino Uno R3 la frecuencia es de 490Hz [9], es decir que para una señal de reloj por medio de esos puertos, se tendría una frecuencia estándar. Si dentro de la solución de salida de reloj se desea obtener una frecuencia diferente a la estándar, se deberá generar una función codificada para que el microcontrolador pueda interactuar con la salida del puerto digital enviando valores altos y bajos con el retraso asociado a la frecuencia deseada.

5.5. ¿Cómo transmitir datos de manera serial por Arduino?

Aunque algunos puertos digitales de Arduino generan salida PWM, no son funcionales para enviar información, por tanto se debe desarrollar una implementación de instrucciones dentro del microcontrolador para que este, por medio de un puerto digital, permita enviar información en trama de 8 bits.

Recordando que los puertos digitales se comportan como switches, es posible enviar tramas binarias con solo controlar la salida digital correspondiente al dato a enviar y enviando el valor asociado a este.

Para lograr lo anterior se deben tener en cuenta las siguientes consideraciones:

- Se define el puerto digital para la transmisión de datos serial.
- La entrada de datos numéricos se realizará por consola. Cabe resaltar que la lectura de datos por consola se realiza tomando de un dato a la vez.
- Los datos numéricos ingresados por consola serán almacenados en una variable String para su posterior manipulación.
- Para el tratamiento de los datos ingresados por consola y su futura conversión a binario y despliegue se requiere de tres funciones que son, conversión de entero a representación en String de binario de 4 bits denominada `intToBinary`, integración de binarios para formar un binario de 8 bits denominada `binary`, y el despliegue o envío de binarios por un puerto digital denominada `sendSerial`.

- Inicialmente el dato numérico que se toma por consola, será almacenado en una variable String, y este será enviado a la función binary; Dentro de la función binary se toma la representación en decimal del dato numérico que está almacenado en la variable String y se envía, por separado, hacia la función intToBinary para que este realice la comparación correspondiente y me reenvíe la representación binaria, en String, del número ingresado. Finalmente esas representaciones binarias de los números tratados son almacenadas en una sola variable creando así un binario de 8 bits el cual representa el dato numérico que se envió por consola.
- Como último paso se tiene reenviar al consolidado de 8bits hacia la función sendSerial, el cual se encarga de coger cada uno de los bits, convertirlos de tipo String a tipo entero y ese valor enviarlo por el puerto digital.

5.5.1. Inclusion de código de comunicacion serial simple entre dos Arduinos Uno R3

Listing 1: Código Arduino Uno R3 como transmisor serial

```
int serialSalida=3;

String dato;

void setup()
{
    Serial.begin(9600);
    pinMode(serialSalida , OUTPUT);
}

void loop() {

    if(Serial.available()>0){

        dato=Serial.read();
        sendSerial(binary(dato));

    }
}
//— funcion que envia trama serial de 8 bits
void sendSerial(String serialData){

    for(int i=0;i<serialData.length();i++){

        int out=String(serialData.charAt(i)).toInt();
```

```

        delay(1000);
        digitalWrite(serialSalida ,out );

    }
}

//—funcion que retorna el binario 8 bits del valor ascci del numero
String binary(String dato)
{
    String res;
    int n=0;

    for(int i=0;i<dato.length();i++){

        n=String(dato.charAt(i)).toInt();
        res+=intToBinary(n);

    }

    return res;
}

//—funcion que convierte un int a string correspondiente a su binario
String intToBinary(int n){
    String binary;
    switch(n){
        case 0:
            binary="0000";
            break;
        case 1:
            binary="0001";
            break;
        case 2:
            binary="0010";
            break;
        case 3:
            binary="0011";
            break;
        case 4:
            binary="0100";
            break;
        case 5:
            binary="0101";
            break;
        case 6:

```

```

        binary="0110";
        break;
    case 7:
        binary="0111";
        break;
    case 8:
        binary="1000";
        break;
    case 9:
        binary="1001";
        break;
    }
    return binary;
}

```

[language=C++, caption=C digo Arduino Uno R3 como receptor serial]

```

int serialEntrada=4;

void setup()
{
    Serial.begin(9600);
    pinMode(serialEntrada , INPUT);
}

void loop() {

    int entrada=digitalRead(serialEntrada);
    Serial.print(entrada);
    delay(1000);

}

```

6. Marco Experimental

7. Resultado

8. Conclusiones

Con la placa de Arduino es posible realizar un sistema de comunicacion serial con un alto grado de calidad en cuanto a la transmision y recepcion de

informacion.

A pesar de que Arduino ofrese multiples herramientas por medio de sus puertos, tambien tiene sus limitaciones, pero en contraparte está el saber diseñar instrucciones que permitan crear mecanismos para buscar la solucion a un problema en especifico.

Una de las grandes limitantes de Arduino es su memoria, por tanto es ideal tener conciencia sobre el manejo de recursos a la hora de diseñar las instrucciones de código. En esta sección .

Referencias

- [1] wikipedia. Medios de transmisión. [Online]. Available: <https://sites.google.com/site/investigacionesitlm/home/1-3-medios-de-transmision>
- [2] xakata. Encriptar: qué es, para qué sirve y cómo cifrar tus archivos. [Online]. Available: <https://www.xataka.com/basics/encriptar-que-sirve-como-cifrar-tus-archivos>
- [3] EcuRed. ¿cuál es la diferencia entre latch y flip flop? [Online]. Available: <https://es.strephonsays.com/what-is-the-difference-between-latch-and-flip-flop>
- [4] V. P. Nelson. ¿cuál es la diferencia entre latch y flip flop? [Online]. Available: <http://mumoaldigitales1.blogspot.com/2010/11/diferencia-entre-los-tipos-de-flip-flop.html>
- [5] ONSemiconductor. 8-bit serial-input/serial or parallel-output shift register with latched3-state outputs. [Online]. Available: <https://pdf1.alldatasheet.com/datasheet-pdf/view/12198/ONSEMI/74HC595.html>
- [6] Arduino. ¿qué es arduino? [Online]. Available: <https://arduino.cl/que-es-arduino>
- [7] xakata. Qué es arduino, cómo funciona y qué puedes hacer con uno. [Online]. Available: <https://www.xataka.com/basics/que-arduino-como-funciona-que-puedes-hacer>
- [8] Arduino. Digital pins. [Online]. Available: <https://docs.arduino.cc/learn/microcontrollers/digital-pins>
- [9] Hetpro. Arduino pwm con arduino uno. [Online]. Available: <https://hetpro-store.com/TUTORIALES/arduino-pwm/>