Universidad Autónoma de Baja California

Facultad de Ingeniería, Arquitectura y Diseño

Ingeniería Electrónica

Telemedicina segura

Como requisito para obtener el título como Ingeniero en Electrónica

Presenta:

López Castillo Luis Eduardo

Ensenada, Baja California Octubre del 2017

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO

TELEMEDICINA SEGURA

TESIS

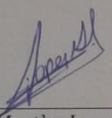
PARA CUBRIR LOS REQUISITOS NECESARIOS PARA OBTENER EL TÍTULO DE

INGENIERO EN ELECTRONICA

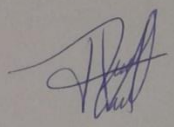
PRESENTA:

LOPEZ CASTILLO LUIS EDUARDO

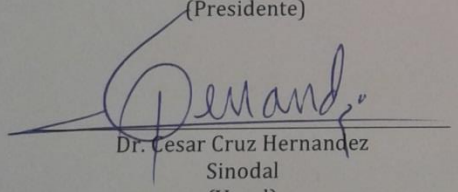
Aprobada por:



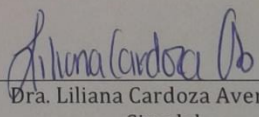
Dra. Rosa Martha Lopez Gutierrez
Directora
(Presidente)



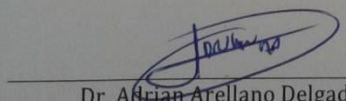
MC. Jose Antonio Michel Macarty
Codirector
(Secretario)



Dr. Cesar Cruz Hernandez
Sinodal
(Vocal)



Dra. Liliana Cardoza Avendano
Sinodal
(Vocal)



Dr. Adrian Arellano Delgado
Sinodal
(Vocal)

Agradecimientos.

A la Dra. Rosa Martha López Gutiérrez y al M.C. José Antonio Michel Macarty como mis asesores de tesis por su apoyo y dedicación que brindaron a lo largo de este proyecto, por todo gracias.

Al Dr. Cesar Cruz Hernandez por brindarme la posibilidad de realizar este trabajo con su apoyo y de la manera más profesional posible.

Al CONACYT a través del proyecto de Investigación No. 166654

Y por ultimo pero aun asi con la misma importancia que todos los anteriores, a mi madre por su apoyo incondicional durante toda mi vida, a mi familia, a mis maestros de la carrera así como mis compañeros que todos ellos me apoyaron para formarme como persona y profesionista, por todo ese apoyo a todos, Gracias.

Índice

Capítulo I

Capitulo II

1.1.- Resumen

Tesis profesional presentada por Luis Eduardo Lopez Castillo como requisito para obtener el titulo en Ingenieria Electronica exponiendo el tema de Envio de información de Telemedicina segura por medio de un sistema caótico que esta formado por un micro-controlador Arduino para poder generar el mapa logístico y dentro del mismo encriptar una señal que es extraida de un documento del tipo .txt con la ayuda de un modulo SD, este sistema se ubica en la parte del emisor en donde se crea el criptograma y envía a través de internet hacia un determinado receptor por lo que este ultimo deberá de incluir un procedimiento similar pero inverso para poder des-encriptar el criptograma y poder recuperar la información que el mensaje original contiene. La comunicación de este sistema de micro-controlador con el modulo SD es del tipo SPI para poder enviar los datos de una manera mas rápida, simple y para simplificar el sistema al menor tamaño. Se utiliza para poder encriptar información de Telemedicina debido a que este tipo de información deberá ser solo del conocimiento del Doctor y su paciente. Con esto se concluye que es posible crear este sistema a partir de un micro-controlador básico como lo es Arduino y es capaz de cumplir con los requerimientos pre-escritos.

1.2.- Introducción:

Los sistemas de protección para la información son esenciales hoy en dia, debido a que una gran parte de esta información vital es manejada via internet para su fácil acceso y manipulación, pero como en todo a una gran ventaja conlleva una desventaja que es el ataque de parte de un intruso que desee esa información para su conveniencia. Debido a esto es un requisito que un sistema de protección tenga la seguridad necesaria contra ataques de este tipo hasta el punto que ni con un sistema computarizado de alta calidad pueda descifrar esa información dispersa en internet.

Cabe señalar que los sistemas deberán de cumplir con las necesidades que desee el cliente y dar la garantía debida. Actualmente estos sistemas pueden llegar a funcionar de distintas maneras tales como protección con claves digitales, sistemas de reconocimientos de rostro entre otras, esas variantes pueden ser aplicadas pero lo mas importante en este caso es que el sistema funcione con la ayuda de la Teoria del caos de la que se tiene relativamente poco uso ya que apenas en los anos 60’s fue cuando se empezó a implementar.

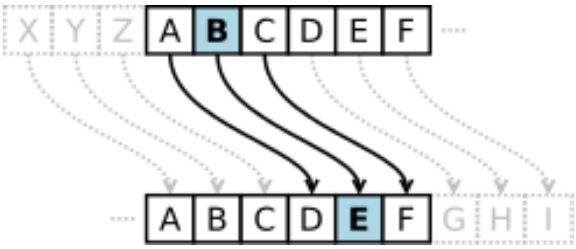
Capitulo II

2.1.- Antecedentes

El cuidado de la información para su posterior envio por un medio publico se ha hecho cuidadosamente por mucho tiempo, tenemos el ejemplo de la antigüedad cuando las tribus guardaban mensajes de guerra importantes en una escitala que llevaba en un pedazo de cuero escrito el mensaje, en este caso la clave era el grosor de la escitala ya que si era de un tamaño incorrecto el mensaje podría leerse de manera distorsionada. Se podría decir que fue uno de los primeros métodos de criptografía y la base de todo lo que existe hoy.

2.1. Escitala espartana, uno de los primeros métodos de encriptamiento de información

En la época de Julio Cesar se usaba un método un poco mas seguro que era el de tener un desplazamiento del abecedario en el sistema de cifrado, lo que hacia era colocar el mensaje por el medio de envio con un desplazamiento el cual consiste en determinar el numero de letras al cual se recorrerán, tomando por ejemplo 4 espacios, con esto si queremos escribir la letra A en el criptograma será la letra E. Un ejemplo mas claro es si deseamos encriptar un mensaje sencillo como “HOLA” en el cifrador de Julio Cesar con desplazamiento de 4 lugares el mensaje seria “LSPE”.



2.2. Cifrado de Julio Cesar, método de encriptamiento.

Anos después en el siglo XIX existio una maquina con la capacidad de encriptar mensajes de gran importancia y de una muy alta privacidad, su nombre era Enigma. Estaba formada por un mecanismo de cifrado rotatorio, que permitía usarla tanto para cifrar como para descifrar mensajes. Existieron varios modelos de este tipo de mecanismo los cuales fueron utilizados en Europa desde inicios del ano de 1920.

Como se dijo anteriormente esta maquina fue utilizada en la guerra y uno de los que requirieron de esta maquina fue las fuerzas militares de Alemania. Su facilidad de manejo y la idea de que era casi imposible descifrar un mensaje generado por esta fueron las principales razones para que el uso de la misma fuera tan solicitada por los militares para enviar sus mensajes. El sistema con el que esta maquina cifraba los mensajes fue estudiado y se logro recuperar mensajes e información que eran importantes y privados de la segunda guerra mundial, algunos dicen que por esta razón fue que esta guerra termino aproximadamente dos anos antes de lo que se esperaba.

Había algunas maquinas equivalentes como Typex que era Britanica y SIGABA que fue Estadounidense, todas ellas eran similares a Enigma. La primera máquina moderna de cifrado rotatorio, de Edward Hebern, era considerablemente menos segura, hecho constatado por William F. Friedman cuando fue ofrecida al gobierno de Estados Unidos.

2.3. Maquina Enigma, sistema de encritamiento

Los sistemas de encriptamiento actuales deben contar con la posibilidad de generar su sistema, o sea, los valores que se sumaran al mensaje de la manera mas rápida y efectiva posible y a su vez poder generar un criptograma que posea un algoritmo publico pero que el numero de claves posibles sea una cantidad tan alta que aun con un sistema de computo muy potente no sea posible poder descifrar la información contenida en el mensaje. [1]

2.2.- Las partes que pueden conformar un sistema criptográfico basico son:

Micro-controlador o micro-procesador:

Este será el encargado de generar el sistema que podría incluir el mapa logístico o método a utilizar. Las especificaciones dependerán de la velocidad que requiera el sistema, en este caso la velocidad dependerá del reloj del sistema además de la frecuencia con que trabajar

Emisor:

Este incluirá un sistema de micro-controlador o micro-procesador con el

sistema que encriptara la información

Receptor:

Sera el mismo sistema solo que en este se realizara el proceso inverso al del emisor

Señal de entrada (mensaje):

Este podrá ser recibido de un medio externo como por ejemplo la señal de un sensor o puede ser generado por otro sistema como uno medico, es importante que los vectores de datos de cada parte como lo es el mensaje y el sistema sean muy similares si es posible iguales refiriéndose a la longitud para que no exista un error a tratar de procesar el criptograma mas de una vez.

Sistema de sensores:

En este caso dependerá del grado de privacidad del sistema, en esta parte se podrán agregar sensores tales como:

- Huella dactilar
- Detector de iris
- Ingreso con clave digital

Actuadores:

En este caso serán los elementos del sistema que se encargaran de realizar una acción con ordenes del procesador, tales como bloquear el sistema físicamente algún puerto de salida o activar alguna alarma en caso de que un intruso este tratando de robar la información que se esta enviando

2.3.- Conceptos generales:

Encriptar, encriptamiento:

Proceso de cifrado de un sistema de seguridad para proteger la información de ataques externos en el momento de realizar un envio publico por internet u otro medio. A la información generada se le conoce como criptograma y es generada por un sistema emisor

Desencriptar, desencriptado:

Proceso inverso al de encriptamiento, en este caso se descifra el criptograma enviado para poder conocer la información tal y como la envió el emisor.

Fractal:

Figura geométrica que posee una dimensión fraccionaria y además tiene un patrón que se repite a pequeñas y gran escala

Atractor:

Figura que se genera al graficar las fases de una señal caotica, cabe destacar que esta figura posee una dimensión fractal (fraccionaria) y son las que describen el comportamiento de un sistema por caos

Micro-controlador:

Es un circuito integrado que posee CPU (Central Processing Unit), memoria RAM (Random Acess Memory) y ROM (Read Only Memory) y esta destinado a realizar una acción en específico por medio de un tipo de programación o lenguaje que tenga por defecto. Tiene puertos de entrada y de salida para uso general y esta destinado a realizar una acción en especifico.

Micro-procesador:

El micro-procesador es el elemento principal que conforma una computadora, es el encargado de ejecutar los programas dentro de la memoria o disco duro y al mismo tiempo también realiza la opción de ejecutar el sistema operativo

Efecto mariposa:

“El pequeño aleteo de una mariposa es una parte del mundo puede generar un huracán en otra”. Frase que trata de describir la teoría del caos y el efecto que pueden llegar a tener pequeñas acciones en un futuro

Cifrado:

Manera de ocultar información para que no sea entendida por cualquier usuario y que solo sea comprensible si se posee la clave necesaria

Sistemas dinámicos:

Un sistema dinámico es un sistema cuyo estado evoluciona con el tiempo. Los sistemas físicos en situación no estacionaria son ejemplos de sistemas dinámicos, pero también existen modelos económicos, matemáticos y de otros tipos que son sistemas abstractos que son, además, sistemas dinámicos.

2.4.- Propósito de encriptar información

El encriptamiento tiene como objetivo poder disenar y desarrollar sistemas capaces de proporcionar seguridad a un sistema para que este sea confidencial y pueda evitar ser interceptado por algún receptor no autorizado al momento de enviar la información por vías públicas. Algunas propiedades de la criptografía son:

- Confidencialidad.

  Es decir, garantiza que la información sea accesible únicamente a personal autorizado. Para conseguirlo utiliza códigos y técnicas de cifrado.

- Integridad.

Es decir garantiza la corrección y completitud de la información.

- Vinculación.

Permite vincular un documento o transacción a una persona o un sistema de gestión criptográfico automatizado. Cuando se trata de una persona, se trata de asegurar su conformidad respecto a esta vinculación de forma que pueda entenderse que la vinculación gestionada incluye el entendimiento de sus implicaciones por la persona. Antiguamente se utilizaba el término "No repudio" que está abandonándose, ya que implica conceptos jurídicos que la tecnología por sí sola no puede resolver. En relación con dicho término se entendía que se proporcionaba protección frente a que alguna de las entidades implicadas en la comunicación, para que no pudiera negar haber participado en toda o parte de la comunicación. Para conseguirlo se puede usar por ejemplo firma digital. En algunos contextos lo que se intenta es justo lo contrario: Poder negar que se ha intervenido en la comunicación. Por ejemplo cuando se usa un servicio de mensajería instantánea y no queremos que se pueda demostrar esa comunicación. Para ello se usan técnicas como el cifrado negable.

- Autenticación.

  Es decir proporciona mecanismos que permiten verificar la identidad del comunicador. Para conseguirlo puede usar por ejemplo función hash criptográfica MAC o protocolo de conocimiento cero.

- Soluciones a problemas de la falta de simultaneidad en la telefirma digital de contratos.

  Para conseguirlo puede usar por ejemplo protocolos de transferencia inconsciente.

2.5.- Características de un sistema criptográfico:

Dependiendo de las necesidades que se establezcan por el usuario existirá una variación de un sistema a otro pero las características básicas de un sistema criptográfico son las siguientes:

- Poseer algoritmo:

  De esta manera se podrá conocer la manera en que la información se cifra y si se tiene acceso a la manipulación del algoritmo se tendrá la capacidad de realizar modificaciones futuras en caso de que el sistema posea una debilidad posible a modificar

- Numero de claves:

  El número de claves posibles que se puedan probar en el sistema antes de que sea corrompido será proporcional a la seguridad del mismo contra ataques.

- Velocidad:
  Esta característica dependerá de la aplicación, si la necesidad de una alta velocidad es indiferente se podrá realizar el sistema con un micro-controlador básico con reloj de unos cuantos Mhz.

- Comodidad:
  Esto se refiere la facilidad con la que se puede usar el sistema además de que tan sencilla es la comprensión

  En pocas palabras las características que definirán un buen sistema básico de encriptamiento será que posea una velocidad promedio que permita realizar las tareas del sistema, su seguridad sea alta ya que ese es el factor mas importante para determinar si es bueno y que su algoritmo pueda modificarse para poder corregir debilidades en caso de que se detecten en el funcionamiento.

2.6.- Implementación:

Como se dijo anteriormente el objetivo de un sistema criptográfico es poder recibir un mensaje o información que un usuario desea enviar mediante una via publica como el internet pero que cuente con la seguridad de que solo podrá ser visible o entendible por el mismo usuario y el receptor, por esta razón se han creado este tipo de sistemas ya que en la actualidad la mayoría de los datos son procesados en medios donde es fácil su extracción.

2.7.- Hipótesis:

Poder generar un sistema que posea las características de un sistema criptográfico, que sea relativamente económico y sea lo más sencillo posible. Además de que la interacción Usuario-Maquina sea la mejor experiencia para una mejor comprensión del sistema. Además de que tenga un algoritmo modificable para poder tener manipulación en caso de que las necesidades del usuario cambien o se requiera de aumentar la seguridad.

2.8.- Grado de seguridad de un sistema criptográfico

Un sistema criptográfico es seguro respecto a una tarea si un adversario con capacidades especiales no puede romper esa seguridad, es decir, el atacante no puede realizar esa tarea específica.

Actualmente para evaluar la seguridad de un sistema criptográfico existe la Politica de revelación, esta establece el grado que la información como algoritmos que forman el sistema son revelados para cualquier usuario, esto tiene sus ventajas y desventajas. Una de las ventajas y quizás la mas importante es que al revelar el algoritmo implementado en el sistema podemos darnos cuenta de algunas de las debilidades de nuestro sistema para poder corregirlas aunque esto a su vez implica que se tendrán un mayor numero de ataques para poder revelar el mensaje que se envía.

Basándose en el principio de Kerckhoffs podemos deducir que nuestro sistema deberá tener como grado de seguridad el número de claves posibles que se puedan utilizar al intentar descifrar o romper el sistema para intervenir. Con esto se puede deducir que si nosotros revelamos el algoritmo usado para nuestro sistema y aun asi es difícil de descifrar será aun más difícil si no difundimos el algoritmo.

2.9.- Clasificación de sistemas

Los sistemas dinámicos se clasifican en 3 tipos

1.- Estables

2.- Inestables

3.-Caoticos

Los sistemas estables tienen la característica que después de un largo tiempo transcurrido tienden a un punto, normalmente siguen con su misma forma. Debido a esto, los sistemas estables junto con sus ecuaciones características, condiciones iniciales, limites, elementos y relación dan a conocer su comportamiento a través del tiempo y podemos saber hacia donde va el atractor

En el caso de los sistemas inestables es diferentes ya que estos no se guian por atractores y nunca llegan hacia un punto

Un sistema caótico es capaz de presentar ambos comportamientos. Un punto importante de estos sistemas es que tienen sus ecuaciones y condiciones iniciales pero si se varia un valor por mas minimo que sea provocara un cambio muy grande en el comportamiento a lo largo del tiempo, al principio parecerá que los sistemas van de la misma forma pero cambiaran mas a lo largo de que transcurre mas tiempo.

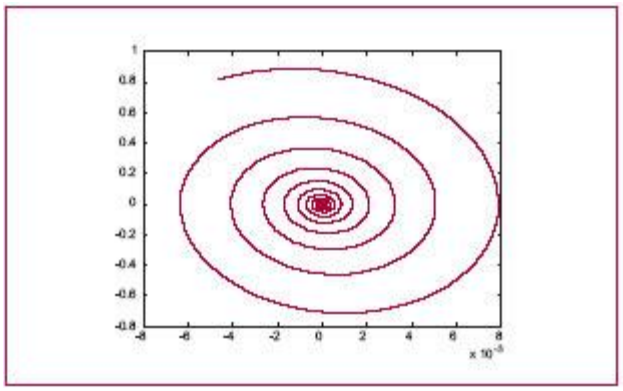
2.10.- Atractores

El comportamiento o movimientos que tiene un sistema caótico normalmente es representado en el espacio de fases, de esta manera es posible conocer un elemento muy importante de este tipo de sistemas que son sus atractores en los cuales podemos observar la trayectoria que va tomando el sistema en el tiempo con respecto a su plano de fases.

Cabe destacar que los atractores pueden presentarse igualmente en los sistemas de punto fijo y de ciclo límite, no son exclusivos de los sistemas caóticos ya que antes de que se hablara de caos se tenía el conocimiento de este tipo de atractores. Ahora se explicaran los diferentes tipos de atractores iniciando con el de punto fijo.

- 2.10.1.- Atractor de punto fijo:

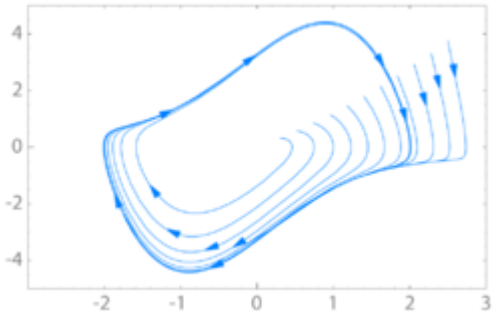
  Corresponde al más simple, el sistema que tenga un atractor de punto fijo tenderá a estabilizarse en un único punto. Un ejemplo común es el péndulo, que tiende al punto en el que el ángulo es nulo respecto a la vertical, debido al rozamiento con el aire. Este atractor si se ve de manera grafica se distinguirá debido a que tendrá un comportamiento más o menos uniforme y después de eso se observara que llega a un punto y de ahí no cambia más aunque el tiempo siga transcurriendo.



. 2.4. Atractor punto fijo

- 2.10.2.- Atractor de ciclo límite o atractor periódico:

  Es el segundo tipo de atractor más sencillo. Este tipo de atractor tiende a mantenerse en un periodo igual para siempre. Como ejemplo, se puede tomar un péndulo alimentado para contrarrestar la fuerza de rozamiento, por lo que oscilaría de lado a lado. Este tipo de atractor también se distingue de manera grafica ya que este después de un tiempo los valores oscilan entre puntos iguales lo cual nos da como resultado una figura que se repite con el tiempo



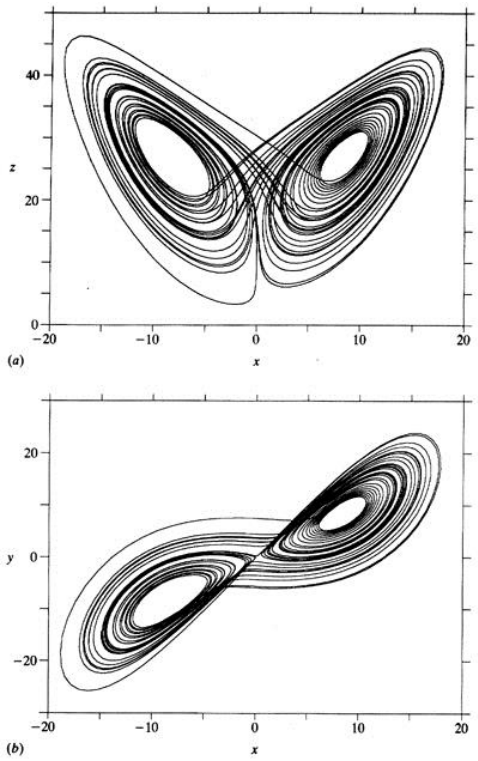
2.5. Atractor ciclo límite

- 2.10.3.- Atractor caótico o atractor extraño :

  Aparece en sistemas no lineales que tienen una gran sensibilidad a las condiciones iniciales, ahora este tipo de atractores gráficamente se observaran como figuras que al principio son un poco extrañas ya que sus valores varian con el tiempo, pero nunca hay una secuencia que se repita si no que son valores determinísticos. A Edward Lorenz se le atribuye sus famosos atractores en forma de alas de mariposa llamados atractor de Lorenz que fue el que inicio con este tipo de investigaciones.

Podemos observar en este caso que dado los ejemplos anteriores con el péndulo en este caso tendrá un impulso constante hacia el péndulo pero este en algunas ocasiones girara en posiciones un poco extrañas que no coincidirán con el impulso

pero que se podrán volver a repetir si se aplican las mismas condiciones iniciales y las mismas ecuaciones. De nuevo, un cambio minimo en sus condiciones generara un cambio total en el sistema



2.6. Atractor caótico o atractor extraño

[2]

2.11.- Aporte de Lorenz

El comienzo de lo que hoy es la teoría del caos nos lleva a los anos de 1950, en estos momentos fueron los inicios de algunas computadoras muy básicas y en la mente de los investigadores había una idea de lo que podrían llegar a ser los sistemas no lineales, ya que al inicio solo se vieron en graficas que demostraban el comportamiento mediante algunos métodos numéricos. Pasaron alrededor de 13 anos lo cual nos lleva al ano de 1963 donde Edward Lorenz trabajaba en unas ecuaciones muy famosas y conocidas mundialmente como ecuaciones de Lorenz las cuales el esperaba que predijeran el tiempo atmosférico, esto lo hizo con un ordenador e ingresando sus ecuaciones para poder ver gráficamente que es lo que estas generaban. El ordenador en el que ingreso sus ecuaciones al realizar los cálculos correspondientes le arrojo a la pantalla lo que hoy en dia es el famoso atractor de Lorenz.

Al ver lo que estaba en la pantalla el supuso que era un error de compilación o alguno en algún valor ingresado en las ecuaciones pero la sorpresa que se llevo es que

siempre le arrojaba el mismo resultado lo cual significaba que no había un error y ese era el resultado correcto. Después de hacer pruebas en el sistema tales como variar las condiciones iniciales y a veces algunas constantes se dio cuenta que al variar muy poco las condiciones iniciales daba como resultado una figura muy diferente. Al observar esto recordó un programa que había escrito para su sistema de meteorología en su computadora en el cual se podían ingresar las condiciones iniciales con 6 decimales pero solo modificaba las últimas 3 y ahí observo que tan delicados son los sistemas aunque sus condiciones sean muy próximas. Casi una década transcurrio para poder darse cuenta y llegar a la conclusión de que en los sistemas meteorológicos no se pueden dar un resultado exacto si no solo uno aproximado debido a que por lo mismo de que sus condiciones iniciales pueden cambiar el sistema cambiara completamente a lo largo del tiempo al igual que el supuesto pronóstico.

Toda la investigación de Lorenz ayudo a que en la década de 1970 fuera un gran avance del caos. En 1971 David Ruelle y Floris Takens propusieron una nueva teoría para la turbulencia de fluidos basada en un atractor extraño. Años después el ecólogo teórico Robert May en 1976 encontró ejemplos de caos en dinámica de poblaciones usando la ecuación logística discreta. A continuación llegó el más sorprendente descubrimiento de todos de la mano de Feigenbaum. Él descubrió que hay un conjunto de leyes universales concretas que diferencian la transición entre el comportamiento regular y el caos, por tanto, es posible que dos sistemas evolucionen hacia un comportamiento caótico igual. [3]

2.11.1.- Ecuaciones de Lorenz

El primer sistema de ecuaciones bien caracterizado que exhibía comportamiento caótico fue el sistema de ecuaciones propuesto por Lorenz:

$$\begin{aligned} \dot{x} &= \sigma(y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz \end{aligned}$$

donde σ es el número de Prandtl (viscosidad/conductividad térmica), r es el número de Rayleigh (John Strutt) (diferencia de temperatura entre base y tope) y b es la razón entre la longitud y altura del sistema.

Lorenz observó dos cosas fundamentales que ocurrían en su ecuación:

1. Cualquier diferencia en las condiciones iniciales antes de los cálculos, incluso infinitesimal, cambiaba de forma drástica los resultados. Tan sólo se podía predecir el sistema por cortos períodos. Llevando eso a la meteorología, suponía lo que se llamó efecto mariposa, hipersensibilidad a las condiciones iniciales.
2. A pesar de lo anterior, la impredecibilidad del sistema, lejos de ser un comportamiento al azar, tenía una curiosa tendencia a evolucionar dentro de

una zona muy concreta del espacio de fases, situando una especie de pseudocentro de gravedad de los comportamientos posibles.

Las ecuaciones de Lorenz fueron propuestas como un modelo muy simplificado de la convección en forma de anillos que parece ocurrir a veces en la atmósfera terrestre. Por ello, las tres magnitudes a las que Lorenz se refiere en su sistema son,

- Razón de rotación del anillo.
- Gradiente de temperatura.
- Desviación de la temperatura respecto a su valor de equilibrio.

Lorenz descubrió que su sistema contenía una dinámica extremadamente errática. Las soluciones oscilaban irregularmente sin llegar a repetirse, aunque lo hacían en una región acotada del espacio de fases. Vio que las trayectorias rondaban siempre alrededor de lo que ahora se define como atractor extraño. El sistema de Lorenz es disipativo. [3]

2.12.- Fractales

Un fractal es un objeto el cual si se observa tiene una estructura que esta fragmentada y al parecer es irregular, su característica principal es que tiene un patrón similar tanto a pequeña como gran escala. En la naturaleza existen muchas estructuras fractales, esto significa que ese objeto debe de poseer la propiedad de que su dimensión métrica fractal es un numero no entero. El término fue propuesto por el matemático Benoît Mandelbrot en 1975 y deriva del latín fractus, que significa quebrado o fracturado

Si bien el término "fractal" es reciente, los objetos hoy denominados fractales eran bien conocidos en matemáticas desde principios del siglo XX. Las maneras más comunes de determinar lo que hoy denominamos dimensión fractal fueron establecidas a principios del siglo XX en el seno de la teoría de la medida

La definición de fractal desarrollada en los años 1970 dio unidad a una serie de ejemplos, algunos de los cuales se remontaban a un siglo atrás. A un objeto geométrico fractal se le atribuyen las siguientes características:

- Es demasiado irregular para ser descrito en términos geométricos tradicionales.
- Es autosimilar, su forma es hecha de copias más pequeñas de la misma figura.

Las copias son similares al todo: misma forma pero diferente tamaño. Ejemplos de autosimilaridad

- Fractales naturales son objetos naturales que se pueden representar con muy buena aproximación mediante fractales matemáticos con autosimilaridad estadística. Los fractales encontrados en la naturaleza se

diferencian de los fractales matemáticos en que los naturales son aproximados o estadísticos y su autosimilaridad se extiende solo a un rango de escalas (por ejemplo, a escala cercana a la atómica su estructura difiere de la estructura macroscópica).

- Conjunto de Mandelbrot es un fractal autosimilar, generado por el conjunto de puntos estables de órbita acotada bajo cierta transformación iterativa no lineal.
- Paisajes fractales, este tipo de fractales generados computacionalmente pueden producir paisajes realistas convincentes.
- Fractales de pinturas, se utilizan para realizar el proceso de decalcomanía.

- Su dimensión de Hausdorff-Besicovitch es estrictamente mayor que su dimensión topológica.
- Se define mediante un simple algoritmo recursivo.

No basta con una sola de estas características para definir un fractal. Por ejemplo, la recta real no se considera un fractal, pues a pesar de ser un objeto autosimilar carece del resto de características exigidas.

Un fractal natural es un elemento de la naturaleza que puede ser descrito mediante la geometría fractal. Las nubes, las montañas, el sistema circulatorio, las líneas costeras o los copos de nieve son fractales naturales. Esta representación es aproximada, pues las propiedades atribuidas a los objetos fractales ideales, como el detalle infinito, tienen límites en el mundo natural.

2.7. Ejemplo de un fractal

La relación básica que posee el caos con los fractales es debido a que cuando se grafica en plano de fases los estados de ecuaciones de caos, por ejemplo las ecuaciones de Lorenz genera lo que ya se vio anteriormente con el nombre de Atractor caótico o Atractor extraño y si se observa este atractor podemos determinar que su dimensión que no pertenece a una entera si no una dimensión fraccionaria o fractal, que es el caso de las figuras que anteriormente se describieron con un patrón que se repite a pequeña y gran escala. [4]

Aplicaciones

El tiempo atmosférico es considerado un sistema dinamico que es muy sensible a los cambios que se pueden llegar a generar en sus variables iniciales, se considera un sistema que es transitivo y tiene orbitas periódicas bastante densas por lo que es un tema en donde claramente se puede aplicar la teoría del caos y trabajar colectivamente. Hasta la actualidad las predicciones que los meteorólogos no proporcionan son relativas y pueden cambiar y no pasar lo que predicen, por lo mismo sigue y seguirá siendo una aproximación.

Al final del siglo XX se ha llegado a poder generar un pronostico con una precisión de entre 80 y 85% en plazos de un día. Los modelos numéricos estudiados en la teoría del caos han dado lugar a generar considerables mejoras en la exactitud de los pronosticos meteorológicos en comparación con las predicciones anteriores, realizadas por medio de métodos subjetivos, en especial para periodos superiores a un día. Actualmente es posible demostrar la confiabilidad de las predicciones específicas para periodos de hasta cinco días gracias a la densidad entre las órbitas periódicas del sistema y se han logrado algunos éxitos en la predicción de variaciones anormales de la temperatura y la pluviosidad para periodos de hasta 30 días.

Antes de la aparición de la Teoría del Caos, se pensaba que para que el tiempo llegara a ser predecido con exactitud newtoniana no era más que una cuestión de introducir más y más variables en un ordenador lo suficientemente potente como para procesarlas. Sin embargo, de unas pocas variables de hace tan sólo unas décadas se ha pasado a considerar cientos de miles de variables sin conseguir la predicibilidad esperada. El clima, como sistema caótico, ha de entenderse como un sistema impredecible dentro de un atractor que le confiere cierto orden a través de las estaciones. Más recientemente se ha probado que el carácter caótico del tiempo atmosférico tiene que ver con las propiedades geométricas del grupo de evolución del sistema climático terrestre, en concreto dicho grupo puede dotarse de la estructura de una variedad de Riemann de dimensión infinita con curvatura negativa, lo cual implica que curvas arbitrariamente cercanas acaban divergiendo en el tiempo. Estos resultados sugieren una imposibilidad práctica de predecir el tiempo atmosférico a medio y largo plazo. El clima es sensible a pequeñas variaciones en las condiciones iniciales y la determinación de las condiciones iniciales con exactitud está abocado al fracaso a causa del Principio de incertidumbre de Heisenberg. Se ha estimado que una predicción a dos meses vista requeriría conocer las condiciones iniciales con una precisión unas 100 mil veces superior a la precisión obtenida por dicha predicción.

2.13.- Sistema de encriptamiento basado en Arduino

Se explicara cada parte del sistema, las características básicas de los elementos que se incluyen y además la razón por la cual se incluyo el elemento en el trabajo descrito

2.13.1.- Arduino Mega 2560 R3

Arduino Mega es una tarjeta de desarrollo open-source construida con un microcontrolador modelo Atmega2560 que posee pines de entradas y salidas (E/S), analógicas y digitales. Esta tarjeta es programada en un entorno de desarrollo que implementa el lenguaje Processing/Wiring. Arduino puede utilizarse en el desarrollo de objetos interactivos autónomos o puede comunicarse a un PC a través del puerto serial (conversión con USB) utilizando lenguajes como Flash, Processing, MaxMSP. Las posibilidades de realizar desarrollos basados en Arduino tienen una muy extensa variedad.

El Arduino Mega tiene 54 pines de entradas/salidas digitales de las cuales se dividen en

- 14 que pueden ser utilizadas como salidas PWM.
- 16 entradas análogas.
- 4 UARTs (puertos serial por hardware)
- 256k de memoria flash.

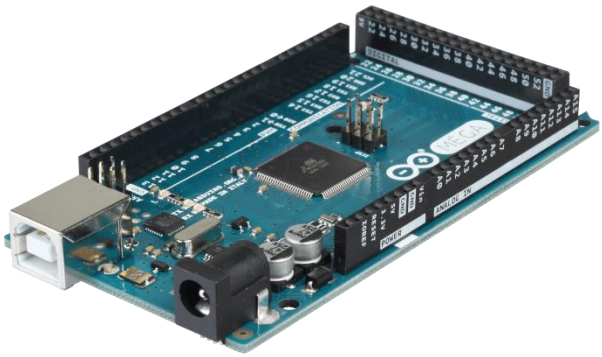
Además como hardware posee

- Cristal oscilador de 16MHz
- conexión USB
- Jack de alimentación
- Conector ICSP
- Botón de reset.

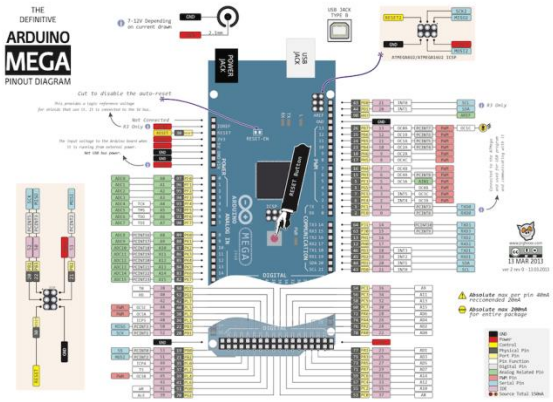
Arduino Mega incorpora todo lo necesario para que el microcontrolador trabaje; simplemente se conecta a un equipo de computo por medio de un cable USB o con una fuente de alimentación externa (9 hasta 12VDC). El Arduino Mega es

compatible con la mayoría de los shields diseñados para Arduino Duemilanove, diecimila o UNO.

Todas estas caracteristicas permiten mayores velocidades de transmisión por el puerto USB y no requiere drivers además ahora cuenta con la capacidad de ser reconocido por el PC como un teclado, mouse, joystick. [5]



2.8 Placa de pruebas del Micro-controlador Arduino



2.9 Puertos de la placa Arduino

La razón del uso del Micro-controlador Arduino Mega es debido a su bajo costo en general ya que en el mercado actual ronda su precio por los $650 pesos que es más barato con respecto a micro-procesadores poco más potentes y que además tiene los

suficientes puertos para tener la posibilidad de agregar elementos externos al sistema sin tener que recurrir a usar otra placa además de la principal

2.13.2.- Interfaz Periférica Serial (SPI)

Interfaz periférica serie (SPI) es un protocolo de datos en serie síncrono utilizado por microcontroladores para comunicarse con uno o más dispositivos periféricos rápidamente en distancias cortas. También se puede utilizar para la comunicación entre dos microcontroladores.

Con una conexión SPI siempre hay un dispositivo maestro (generalmente un microcontrolador) que controla los dispositivos periféricos. Normalmente hay tres líneas comunes a todos los dispositivos:

MISO (Master In Slave Out) - La línea Slave para enviar datos al maestro,

MOSI (Master Out Slave In) - La línea maestra para enviar datos a los periféricos,

SCK (Serial Clock) - Los pulsos de reloj que sincronizan la transmisión de datos generada por el maestro

Y una línea específica para cada dispositivo:

SS (Slave Select): el pin de cada dispositivo que el maestro puede usar para habilitar y deshabilitar dispositivos específicos.

Cuando el pin de Selección de esclavo de un dispositivo está bajo, se comunica con el maestro. Cuando es alto, ignora al maestro. Esto le permite tener varios dispositivos SPI compartiendo las mismas líneas MISO, MOSI y CLK.

Cuando se desea conectar un dispositivo como esclavo a la modalidad SPI se debe tomar en cuenta cual es la velocidad máxima que soporta, esto es controlado por el primer parámetro en SPI Settings. Si utiliza un chip de 15 MHz, utilice 15000000. Arduino utilizará automáticamente la mejor velocidad que sea igual o menor que el número que utiliza con SPISettings.

Además debe tener conocimiento si los datos se desplazan primero en Bit más significativo (MSB) o Bit menos significativo (LSB). Esto se controla mediante el

segundo parámetro SPISettings, ya sea MSBFIRST o LSBFIRST. La mayoría de los chips SPI usan el primer orden de datos MSB.

Otro parámetro a tomar en cuenta es si el reloj de datos está inactivo cuando está en alto o bajo o si hay muestras en el flanco ascendente o descendente de los impulsos de reloj, esto debido a que estos modos son controlados por el tercer parámetro en SPISettings.

El estándar SPI está suelto y cada dispositivo lo implementa un poco diferente. Esto significa que debe prestar especial atención a la hoja de datos del dispositivo al escribir su código.

En general, hay cuatro modos de transmisión. Estos modos controlan si los datos se desplazan hacia adentro y hacia afuera en el flanco ascendente o descendente de la señal de reloj de datos (llamada la fase de reloj) y si el reloj está inactivo cuando es alto o bajo (llamado polaridad del reloj). Los cuatro modos combinan la polaridad y la fase según esta tabla:

Polaridad del reloj en modo (CPOL) Fase del reloj (CPHA) Captura de datos del borde de salida

SPI_MODE0 0 0 Caída en aumento

SPI_MODE1 0 1 Rising Falling

SPI_MODE2 1 0 Rising Falling

SPI_MODE3 1 1 Caída en aumento

Una vez que se tengan los parámetros SPI, use SPI.beginTransaction () para comenzar a usar el puerto SPI. El puerto SPI se configurará con todos sus ajustes. La forma más simple y eficiente de utilizar SPISettings es directamente dentro de SPI.beginTransaction (). Por ejemplo:

SPI.beginTransaction (SPISettings (14000000, MSBFIRST, SPI_MODE0));

Si otras bibliotecas usan SPI de interrupciones, se les impedirá acceder a SPI hasta que llame a SPI.endTransaction (). La configuración de SPI se aplica al inicio de la transacción y SPI.endTransaction () no cambia la configuración de SPI. A menos que usted, o alguna biblioteca, comience a iniciar Transacción una segunda vez, se mantiene la configuración. Debe intentar minimizar el tiempo entre antes de llamar a

SPI.endTransaction (), para una mejor compatibilidad si su programa se utiliza junto con otras bibliotecas que utilizan SPI.

Con la mayoría de los dispositivos SPI, después de SPI.beginTransaction (), escribirá el slave select pin LOW, llame a SPI.transfer () cualquier número de veces para transferir datos, luego escriba el alfiler SS HIGH y finalmente llame a SPI.endTransaction ().

2.13.2.1.- Conexiones SPI para diferentes Arduinos

La tabla siguiente muestra en qué pines las líneas SPI están divididas en las diferentes tarjetas de Arduino:

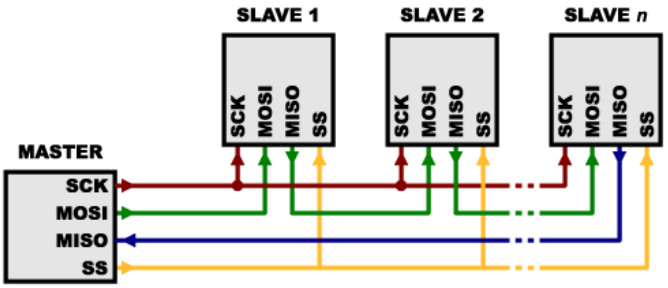
Arduino / Genuino Board	MOSI	MISO	SCK	SS (slave)	SS (master)	Level
Uno or Duemilanove	11 or ICSP-4	12 or ICSP-1	13 or ICSP-3	10	-	5V
Mega1280 or Mega2560	51 or ICSP-4	50 or ICSP-1	52 or ICSP-3	53	-	5V
Leonardo	ICSP-4	ICSP-1	ICSP-3	-	-	5V
Due	ICSP-4	ICSP-1	ICSP-3	-	4, 10, 52	3,3V
Zero	ICSP-4	ICSP-1	ICSP-3	-	-	3,3V
101	11 or ICSP-4	12 or ICSP-1	13 or ICSP-3	10	10	3,3V
MKR1000	8	10	9	-	-	3,3V

2.10. Conexiones SPI en diferentes placas Arduino

La comunicación entre el microcontrolador y la tarjeta SD utiliza SPI, que tiene lugar en las clavijas digitales 11, 12 y 13 (en la mayoría de las tarjetas Arduino) o 50, 51 y 52 (Arduino Mega). Además, se debe utilizar otro pin para seleccionar la tarjeta SD. Éste puede ser el pin de hardware SS - pin 10 (en la mayoría de las tarjetas Arduino) o el pin 53 (en el Mega) - u otro pin especificado en la llamada a SD.begin (). Debe de tenerse en cuenta que incluso si no se utiliza el pin SS del hardware, debe dejarse como una salida o la biblioteca SD no funcionará. Diferentes tablas utilizan diferentes clavijas para

esta funcionalidad, así que asegúrese de haber seleccionado el pin correcto en SD.begin ().

No todas las funciones se enumeran en la página principal de la biblioteca SD, ya que forman parte de las funciones de utilidad de la biblioteca.



2.11. Ejemplo de aplicación de la conexión SPI

Para que este sistema trabaje necesitara de formatear la tarjeta SD y dependiendo del modulo tendrá que ser en FAT 16 o 32. [6]

2.13.- Formateo / Preparación de la tarjeta

(Nota: siempre que se refiera a la tarjeta SD, significa SD y tamaños microSD, así como los formatos SD y SDHD)

La mayoría de las tarjetas SD funcionan bien de la caja, pero es posible que tenga una que se utilizó en una computadora o cámara y no puede ser leída por la biblioteca SD. El formateo de la tarjeta creará un sistema de archivos al que Arduino podrá leer y escribir.

ADVERTENCIA: No es debe dar formato a tarjetas SD con frecuencia, ya que acorta su vida útil.

Necesitará un lector SD y una computadora para formatear la tarjeta SD. La biblioteca admite los sistemas de archivos FAT16 y FAT32, pero utiliza FAT16 cuando sea posible. El proceso de formato es bastante sencillo. A continuacion se describe para diferentes sistemas operativos

2.13.1.- Formateo de SD para diferentes sistemas operativos

Windows: haga clic derecho en el directorio de su tarjeta y elija "Formato" en el menú desplegable. Asegúrese de elegir FAT como el sistema de archivos.

OSX: abra la utilidad de disco (situada en Aplicaciones> Utilidades). Elija la tarjeta, haga clic en la ficha borrar, seleccione MS-DOS (FAT) como el formato y haga clic en Borrar. Nota: OSX coloca una serie de archivos "ocultos" en el dispositivo cuando formatea una unidad. Para formatear un coche SD sin los archivos adicionales en OSX.

Linux: Con una tarjeta SD insertada, abra una ventana de terminal. En el indicador, escriba df y presione enter. Las ventanas informarán el nombre del dispositivo de su tarjeta SD, debería parecer algo como / dev / sdb1. Desmonte la tarjeta SD, pero déjela en el ordenador. Escriba sudo mkdosfs -F 16 / dev / sdb1, reemplazando el nombre del dispositivo por el suyo. Retire la tarjeta SD y reemplácela para verificar que funciona.

Nombre de archivo

Los sistemas de archivos FAT tienen una limitación cuando se trata de convenciones de nomenclatura. Debe utilizar el formato 8.3 para que los nombres de archivo se parezcan a "NAME001.EXT", donde "NAME001" es una cadena de 8 caracteres o menos y "EXT" es una extensión de 3 caracteres. Las personas usan comúnmente las extensiones .TXT y .LOG. Es posible tener un nombre de archivo más corto (por ejemplo, mydata.txt o time.log), pero no puede utilizar nombres de archivo más largos. [7]

2.12. Memoria Micro SD y Adaptador

Ahora se describirá un poco el uso de algunos comandos del modulo SD para cuando ya se inicie la programación del sistema en código C

2.13.2.- Apertura / cierre de archivos (Código para el modulo SD)

Cuando utiliza file.write (), no escribe en la tarjeta hasta que no haga flush () o close (). Siempre que abra un archivo, asegúrese de cerrarlo para guardar sus datos.

Cabe destacar que algunas versiones de modulos SD permiten tener múltiples archivos abiertos.

2.13.3.- El sistema de archivos FAT16

El primer sistema de archivos en ser utilizado en un sistema operativo de Microsoft fue el sistema FAT, que utiliza una tabla de asignación de archivos. La tabla de asignación

de archivos es en realidad un índice que crea una lista de contenidos del disco para grabar la ubicación de los archivos que éste posee. Ya que los bloques que conforman un archivo no siempre se almacenan en el disco en forma contigua (un fenómeno llamado fragmentación), la tabla de asignación permite que se mantenga la estructura del sistema de archivos mediante la creación de vínculos a los bloques que conforman el archivo. El sistema FAT es un sistema de 16 bits que permite la identificación de archivos por un nombre de hasta 8 caracteres y tres extensiones de caracteres. Es por esto que el sistema se denomina FAT16.

Para mejorar esto, la versión original de Windows 95 (que usa el sistema FAT16) se lanzó al mercado con una administración FAT mejorada en la forma del sistema VFAT (Virtual FAT [FAT Virtual]). VFAT es un sistema de 32 bits que permite nombres de archivos de hasta 255 caracteres de longitud. Sin embargo, los programadores tenían que asegurar una compatibilidad directa para que los entornos (DOS) de 16 bits aún pudieran acceder a estos archivos. Por ende, la solución fue asignar un nombre para cada sistema. Por esta razón se pueden usar nombres extensos de archivos en Windows 95 y, aun así, acceder a ellos en DOS.[8]

2.13.4.- Sistema de archivos FAT32

Aunque el VFAT era un sistema inteligente, no afrontaba las limitaciones de FAT16. Como resultado, surgió un nuevo sistema de archivos en Windows 95 OSR2 (el cual no sólo contaba con una mejor administración FAT como fue el caso de VFAT). Este sistema de archivos, denominado FAT32 utiliza valores de 32 bits para las entradas FAT. De hecho, sólo se utilizan 28 bits, ya que 4 bits se reservan para su uso en el futuro.

Cuando surgió el sistema de archivos FAT32, el máximo número de clúster por partición aumentó de 65535 a 268.435.455 (2^{28}-1). Por lo tanto, FAT32 permite particiones mucho más grandes (hasta 8 terabytes). Aunque en teoría, el tamaño máximo de una partición FAT32 es de 8 TB, Microsoft lo redujo, voluntariamente, a 32 GB en los sistemas 9x de Windows para promover NTFS. Ya que una partición FAT32 puede contener muchos clúster más que una partición FAT16, es posible reducir significativamente el tamaño de los clúster y, así, limitar también el espacio desperdiciado del disco. Por ejemplo, con una partición de 2 GB, es posible usar clúster de 4KB con sistemas FAT32 (en lugar de clúster de 32KB con sistemas FAT16), que reducen el espacio desperdiciado por un factor de 8.

El intercambio radica en que FAT32 no es compatible con las versiones de Windows previas al OEM Service Release 2. Un sistema que arranque con una versión anterior simplemente no verá este tipo de particiones. Asimismo, las utilidades de administración de un disco de 16 bits, como ser versiones antiguas de Norton Utilities, ya no funcionarán correctamente. En términos de

realización, el uso de un sistema FAT32 en lugar de un sistema FAT16 tendrá como resultado una leve mejora, de aproximadamente 5%, en el rendimiento.

Tamaño del clúster	Sistema de archivos FAT16	Sistema de archivos FAT32 (en teoría)
512 bytes	32 MB	64 MB
1 KB	64 MB	128 MB
2 KB	128 MB	256 MB
4 KB	256 MB	8 GB (1 TB)
8 KB	512 MB	16 GB (2 TB)
16 KB	1 GB	32 GB (4 TB)
32 KB	2 GB	2 GB (8 TB)

2.15. Tabla de comparación entre FAT16 y FAT32

Los puertos serie son la forma principal de comunicar una placa Arduino con un ordenador. Gracias al puerto serie podemos, por ejemplo, mover el ratón o simular la escritura de un usuario en el teclado, enviar correos con alertas, controlar un robot realizando los cálculos en el ordenador, encender o apagar un dispositivo desde una página Web a través de Internet, o desde una aplicación móvil a través de Bluetooth.

Existen un sin fin de posibilidades en las que se requiere el empleo del puerto serie. Por tanto el puerto serie es un componente fundamental de una gran cantidad de proyectos de Arduino, y es uno de los elementos básicos que debemos aprender para poder sacar todo el potencial de Arduino.[8]

2.14.- Software Arduino

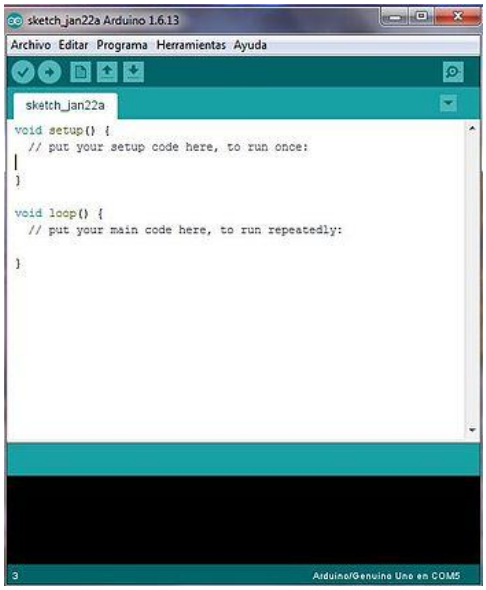
El software de Arduino es donde se programara la placa del micro-controlador y se hablara un poco de su informacion

Arduino es una compañía de hardware libre y una comunidad tecnológica que diseña y manufactura placas computadora de desarrollo de hardware y software, compuesta respectivamente por circuitos impresos que integran un micro controlador y un entorno de desarrollo (IDE), en donde se programa cada placa.

Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios. Toda la plataforma, tanto para sus componentes de hardware como de software, son liberados con licencia de código abierto que permite libertad de acceso a ellos.

El hardware consiste en una placa de circuito impreso con un micro controlador, usualmente Atmel AVR, puertos digitales y analógicos de entrada/salida, los cuales pueden conectarse a placas de expansión (shields), que amplían las características de funcionamiento de la placa Arduino. Asimismo, posee un puerto de conexión USB desde donde se puede alimentar la placa y establecer comunicación con el sistema de computo.

Por otro lado, el software consiste en un entorno de desarrollo (IDE) basado en el entorno de processing y lenguaje de programación basado en Wiring, así como en el cargador de arranque (bootloader) que es ejecutado en la placa. El microcontrolador de la placa se programa mediante un computador, usando una comunicación serial mediante un convertidor de niveles RS-232 a TTL serial.[9]



2.13. Ejemplo de pantalla Software Arduino

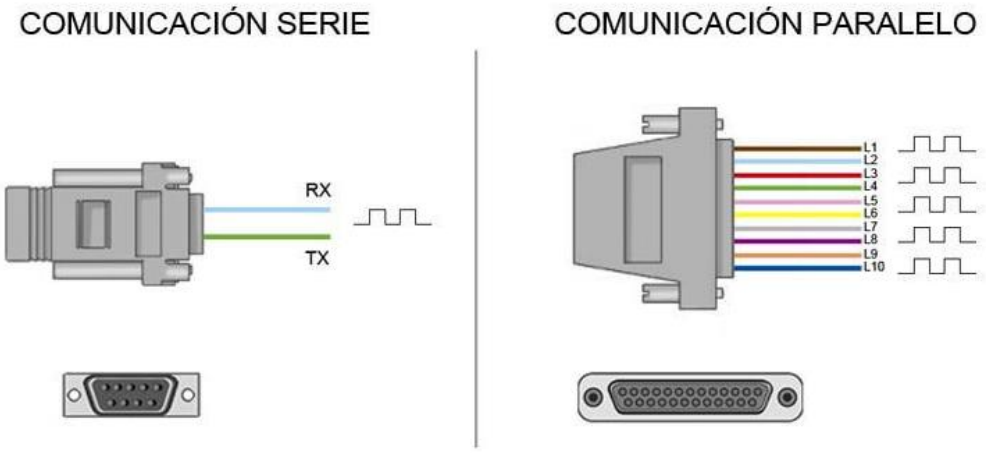
2.14. Software Arduino

2.15.- El puerto serie

Un puerto es el nombre genérico con que denominamos a los interfaces, físicos o virtuales, que permiten la comunicación entre dos ordenadores o dispositivos**.**

Un puerto serie envía la información mediante una secuencia de bits. Para ello se necesitan al menos dos conectores para realizar la comunicación de datos, RX (recepción) y TX (transmisión). No obstante, pueden existir otros conductores para referencia de tensión, sincronismo de reloj, etc.

Por el contrario, un puerto paralelo envía la información mediante múltiples canales de forma simultánea. Para ello necesita un número superior de conductores de comunicación, que varían en función del tipo de puerto. Igualmente existe la posibilidad de conductores adicionales además de los de comunicación.



2.16. Conectores de conexión Serie y Paralelo

Históricamente ambos tipos de puertos han convivido en los ordenadores, empleándose los puertos paralelos en aquellas aplicaciones que requerían la transmisión de mayores volúmenes de datos. Sin embargo, a medida que los

procesadores se hicieron más rápidos los puertos de serie fueron desplazando progresivamente a los puertos paralelos en la mayoría de aplicaciones.

Un ordenador convencional dispone de varios puertos de serie. Los más conocidos son el popular USB (universal serial port) y el ya casi olvidado RS-232 (el de los antiguos ratones). Sin embargo, dentro del ámbito de la informática y automatización existen una gran cantidad adicional de tipos de puertos serie, como por ejemplo el RS-485, I2C, SPI, Serial Ata, Pcie Express, Ethernet o FireWire, entre otros.

En ocasiones se podrá observar referirse a los puertos de serie como UART. La UART (universally asynchronous receiver/transmitter) es una unidad que incorporan ciertos procesadores, encargada de realiza la conversión de los datos a una secuencia de bits y transmitirlos o recibirlos a una velocidad determinada.

Por otro lado, también existe el término TTL(transistor-transistor logic). Esto significa que la comunicación se realiza mediante variaciones en la señal entre 0V y Vcc (donde Vcc suele ser 3.3V o 5V). Por el contrario, otros sistemas de transmisión emplean variaciones de voltaje de -Vcc a +Vcc (por ejemplo, los puertos RS-232 típicamente varían entre -13V a 13V).

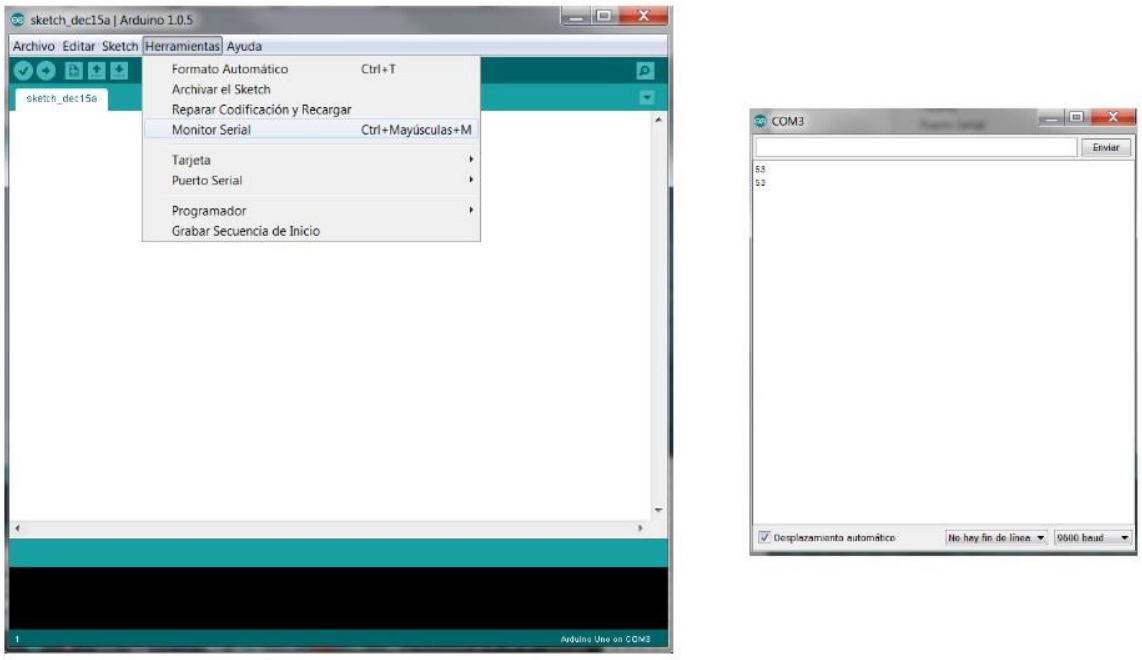
Arduino y puerto serie

Prácticamente todas las placas Arduino disponen al menos de una unidad UART. Las placas Arduino UNO y Mini Pro disponen de una unidad UART que operan a nivel TTL 0V / 5V, por lo que son directamente compatibles con la conexión USB. Por su parte, Arduino Mega y Arduino Due disponen de 4 unidades UART TTL 0V / 5V.

Los puertos serie están físicamente unidos a distintos pines de la placa Arduino. Lógicamente, mientras usamos los puertos de serie no podemos usar como entradas o salidas digitales los pines asociados con el puerto serie en uso.

En Arduino UNO y Mini Pro los pines empleados son 0 (RX) y 1 (TX). En el caso de Arduino Mega y Arduino Due, que tienen cuatro puertos de serie, el puerto serie 0 está conectado a los pines 0 (RX) y 1 (TX), el puerto serie 1 a los pines 19 (RX) y 18 (TX) el puerto serie 2 a los pines 17 (RX) y 16 (TX), y el puerto serie 3 a los pines 15 (RX) y 14 (TX).

Muchos modelos de placas Arduino disponen de un conector USB o Micro USB conectado a uno de los puertos de serie, lo que simplifica el proceso de conexión con un ordenador. Sin embargo algunas placas, como por ejemplo la Mini Pro, prescinden de este conector por lo que la única forma de conectarse a las mismas es directamente a través de los pines correspondientes.



2.17. Herramienta de monitor Serie en Software Arduino

Capitulo III

3.1.- Funcionamiento del sistema

Micro-controlador

En el micro-controlador que es de la marca Arduino se redacto un programa que género el mapa logístico basándose en la siguiente ecuación:

$$x_{n+1} = r x_n (1 - x_n)$$

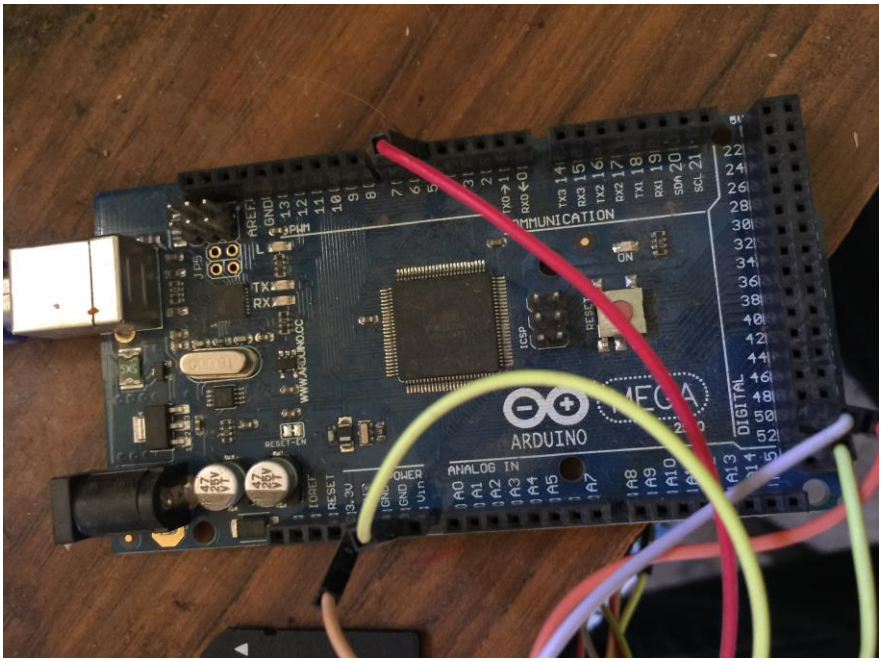
El cual debido a limitantes en el sistema de memoria y espacio se detuvo en 950 iteraciones por que el vector si se declaraba de una mayor extensión el programa al compilar arrojaba un error.

Se asigno el nombre de XX1[tt] a la variable que era compuesta por el sistema al cual se le sumaria el mensaje, en este caso el método para generar el criptograma seria la simple suma del sistema mas el mensaje al ir sumando uno por uno cada elemento del vector hasta que se completara el ciclo. Esos valores serian arrojados por el monitor serial para que el usuario pueda conocerlos y además se agrego la opción para que conforme los datos fueran desplegados se graficaran a través del tiempo y se pudiera observar el criptograma. El sistema graficado fue el siguiente:

Algunos de los problemas que se presentaron al iniciar con el trabajo del código fue el tipo de variables con el que se trabajarían y además el tipo de declaración que necesitaban para poder trabajar correctamente con las iteraciones. Despues de varias pruebas en las que se probaron varios tipos de declaraciones se concluyó el tipo de variable que se utilizo fue flotante para la mayoría de los casos en la gran parte de los resultados contenían al menos 2 digitos después del punto

3.2.- Descripción del sistema

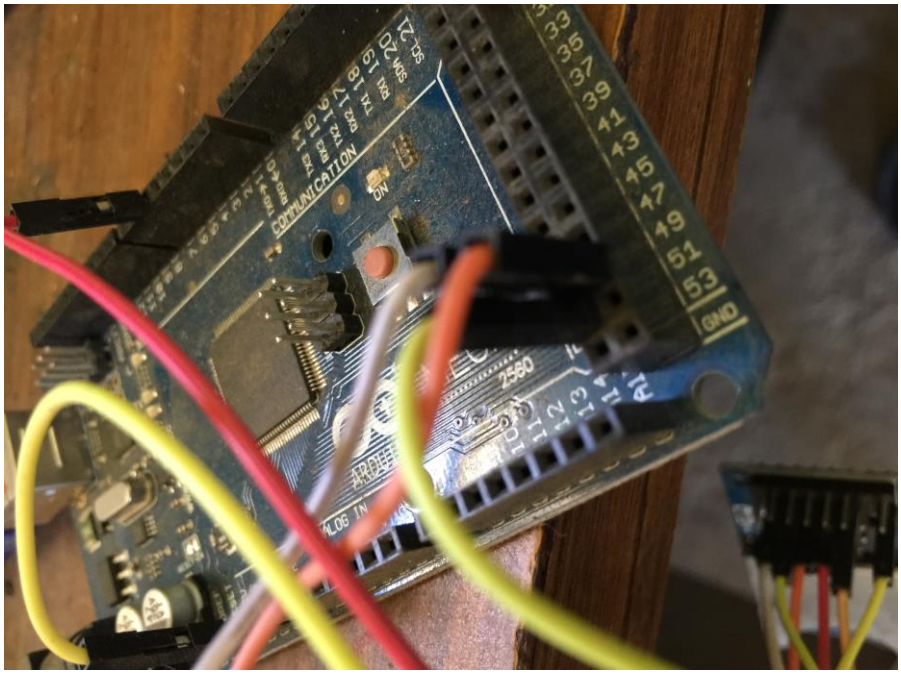
Para comenzar a utilizar el sistema de encriptamiento por medio del micro-controlador Arduino será necesario tener la alimentación conectada a la placa de puebas mediante un cable USB conectado a la computadora o un cargador de 6-12 V diseñado con el conector de alimentación de la placa.



3.1. Placa de pruebas Arduino, conectores de Alimentación (Izquierda)

Además debemos de dar una revisión a las conexiones físicas de los cables a la placa y hacia el modulo SD ya que una mala conexión o un falso contacto no permitirá que el sistema arranque.

3.3.- Las conexiones hacia el modulo son las siguientes:



3.2. Vista de puertos de comunicación MOSI, MISO y SCK

Puerto- Pin

SCK- 52

MOSI- 51

MISO- 50

CS- 5

GND- GND

VCC-5v o 3.3v

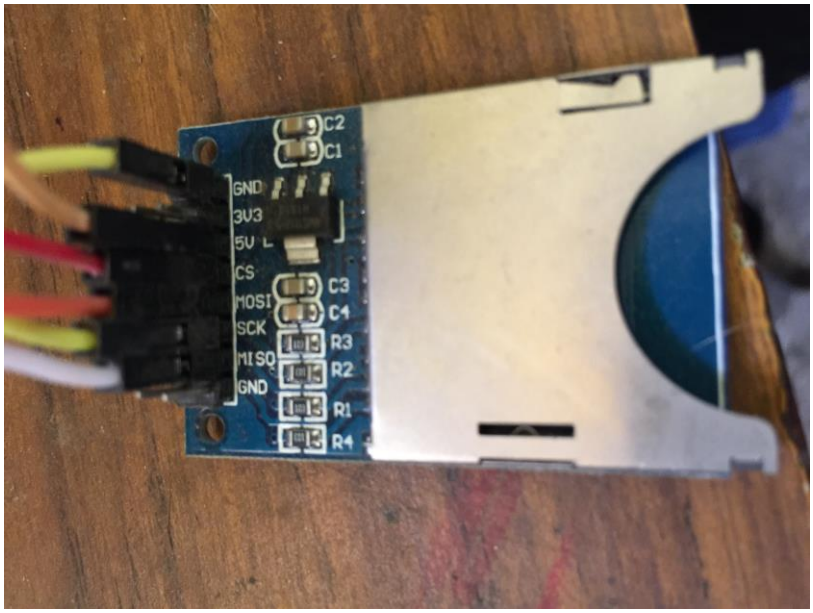
NOTA:

Las conexiones anteriores no están sujetas a cambios debido a que el modulo SD y el Micro-controlador Arduino se comunican mediante conexión SPI la cual ya tiene

destinados puertos por defecto que son los que están descritos anteriormente. El modulo SD no se podrá comunicar si se coloca en otros pines.

El modulo SD y sus pines de conexión se muestran en la siguiente imagen:

3.3. Memoria microSD utilizada con adaptador



3.4. Vista modulo SD para Arduino

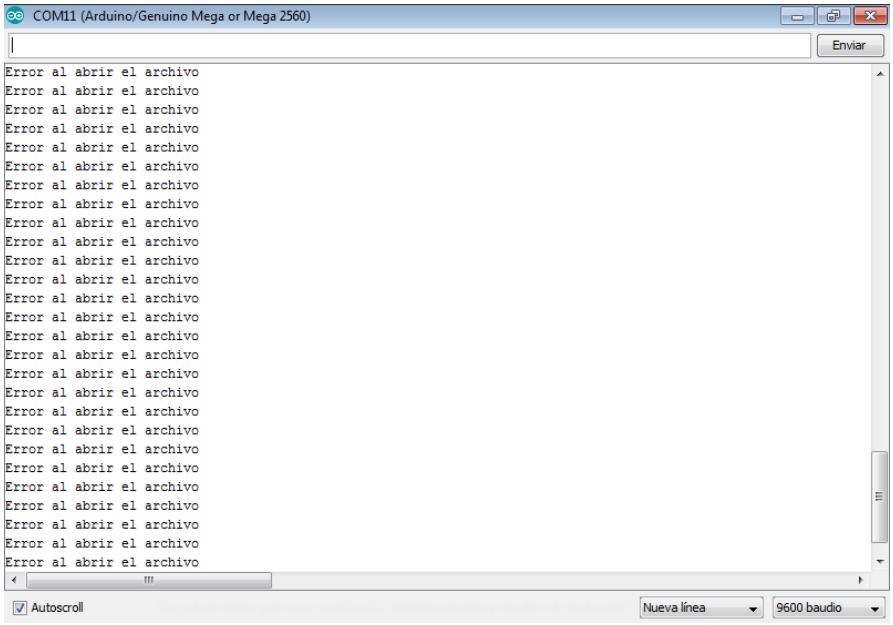
El modulo posee dos pines para conexión en cada puerto, asi que la conexión se puede realizar en el que se desee

Cuando se vaya a iniciar a trabajar verificar que la memoria este bien insertada en el modulo y sea de la capacidad soportada del modulo, teniendo como un máximo una capacidad de 64gb.

ADVERTENCIA:

El modulo SD provee la posibilidad de alimentación de 3.3v y de 5v, por ninguna razón se deberán de mezclar los pines como por ejemplo alimentar con 5v y colocarlo en el puerto de 3.3v del modulo o viceversa ya que puede fallar e incluso quemar el modulo

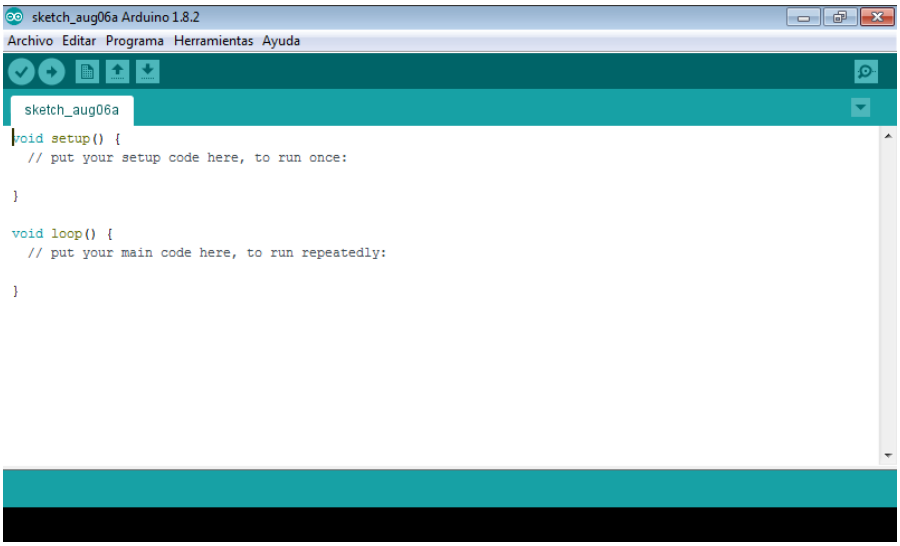
En caso de tener mal conectado algún cable de comunicación e incluso de alimentación el programa enviara un mensaje de error como el de la siguiente imagen por lo que requerirá de dar una revisión visual para poder determinar que es lo que genera ese error. Para conocer un poco mas de los errores del sistema revisar la parte de errores comunes al usar el sistema.



3.5. Vista mensaje de error del sistema

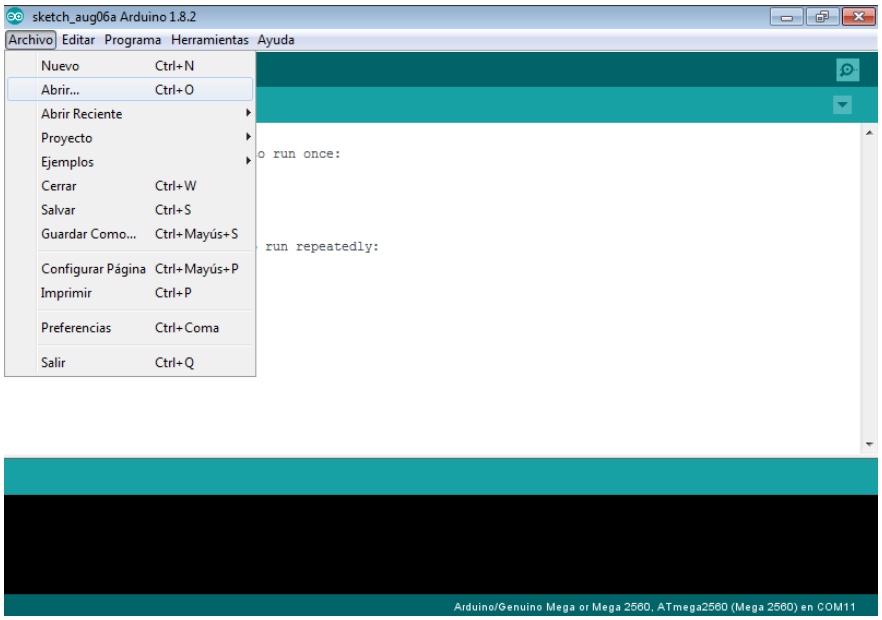
3.4.- Cargar archivo

Una vez que ya se ha corroborado las conexiones físicas del sistema se procederá a abrir el Software Arduino y se necesitara el archivo .ino con el código necesario para cargar en el emisor.



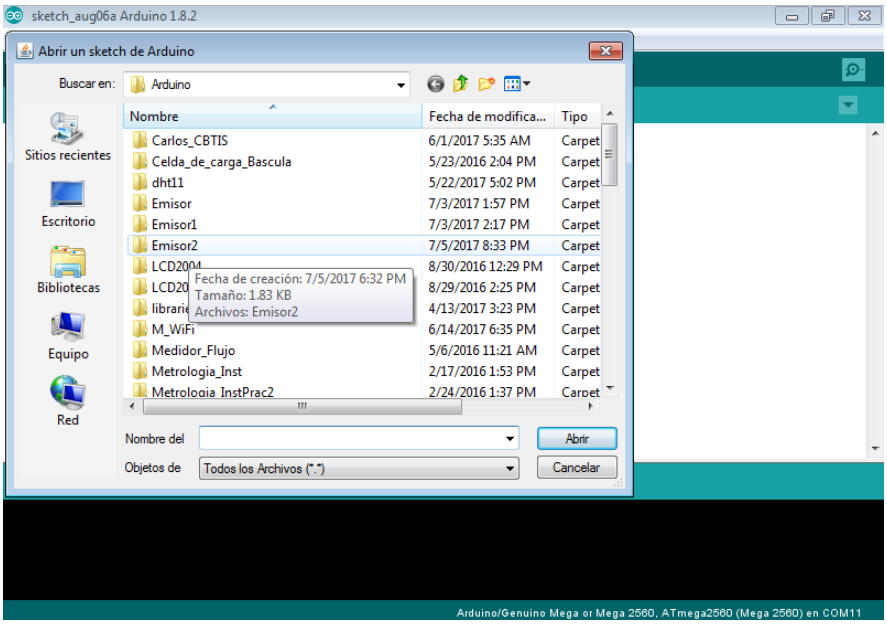
3.6. Software Arduino al ingresar con un nuevo código en blanco

El archivo con el código necesario para la parte del emisor tiene el nombre de Emisor2 y es el que se subirá a la placa para comprobar su correcto funcionamiento. La carga del código se realizara de la siguiente manera:



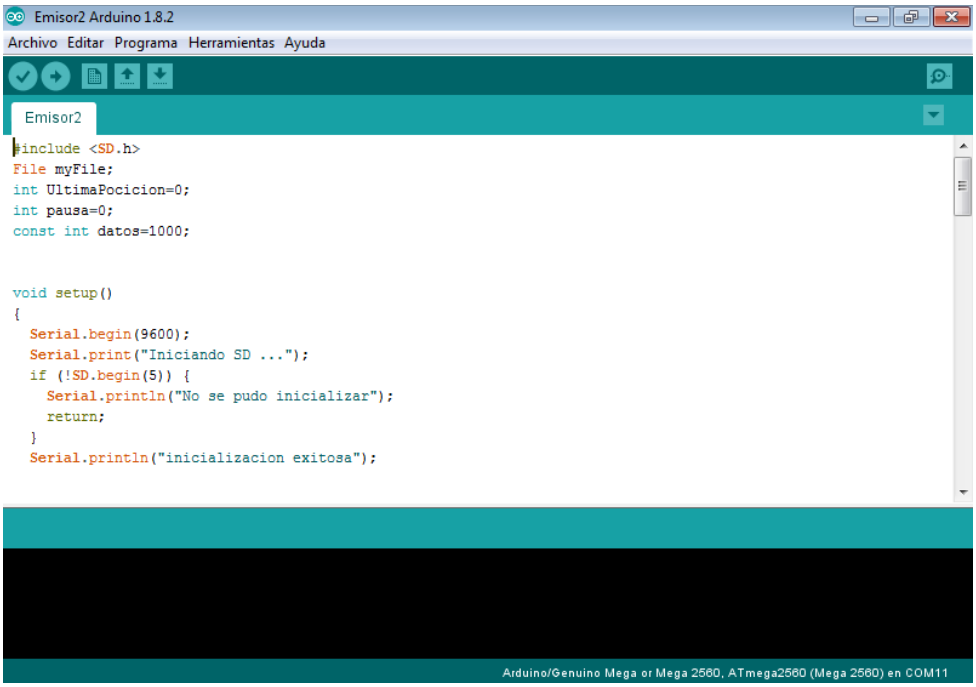
3.7. Abrir archivo del emisor

Se selecciona la parte llamada “Archivo” después se escoge “Abrir”



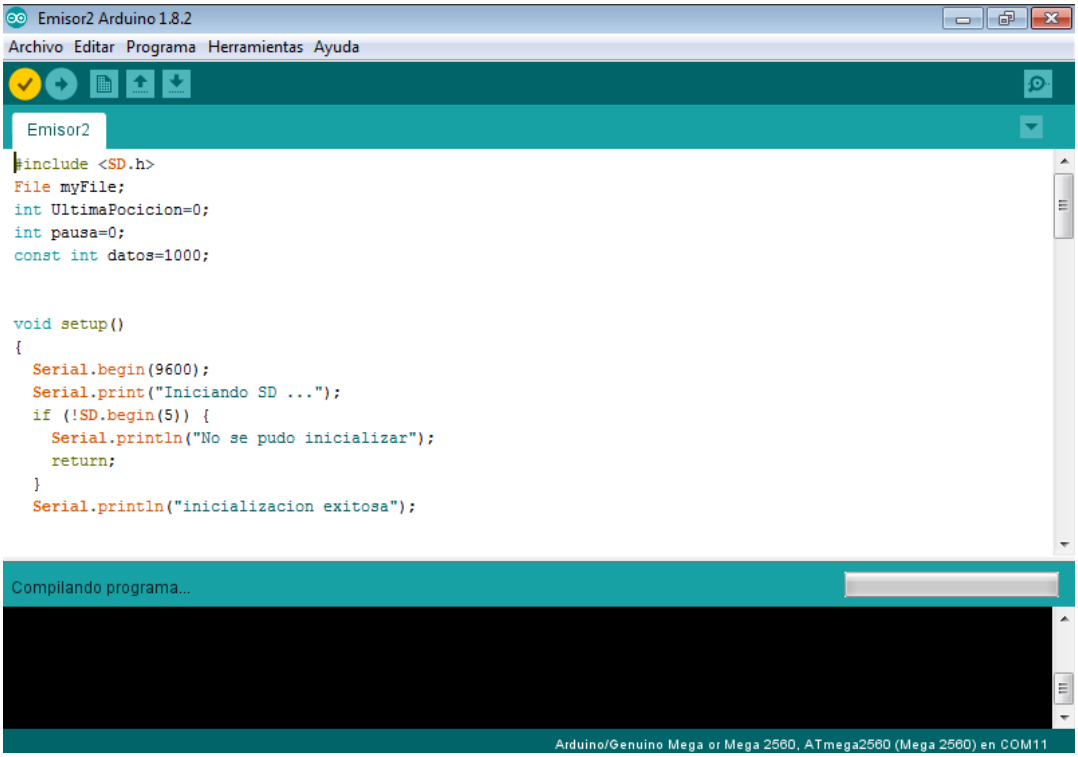
3.8. Búsqueda de código para el sistema emisor

Se busca el archivo con el nombre indicado y se procede a abrir el código.



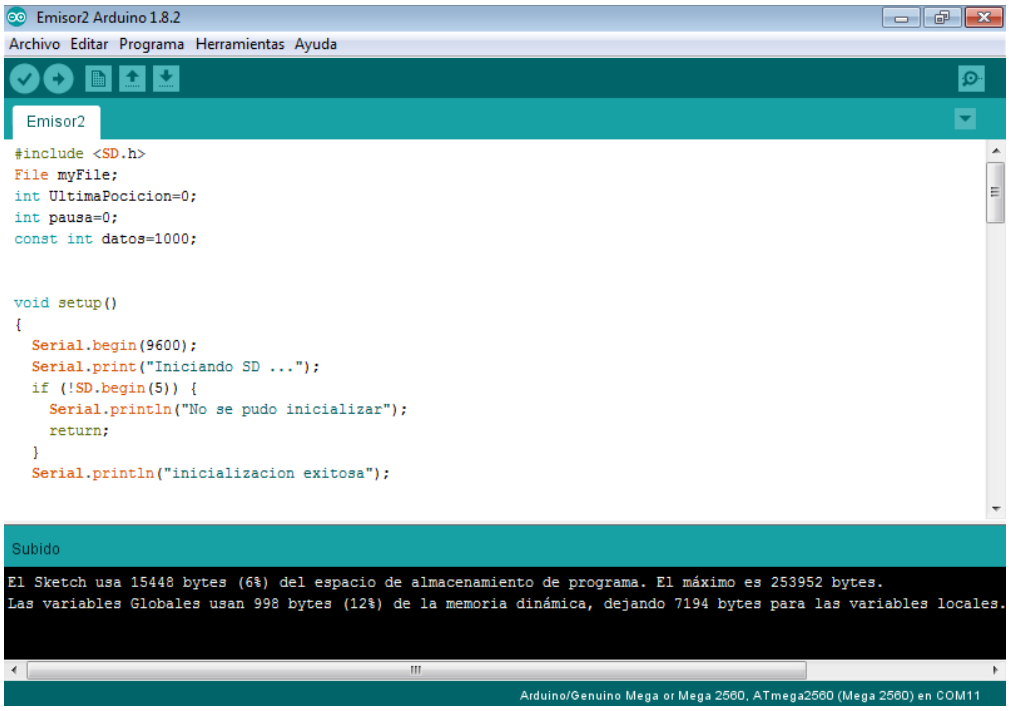
3.9. Código del sistema emisor

Una vez abierto el código se procede a realizar la carga a la placa de pruebas para compilar y ejecutar, para eso se realiza lo siguiente:



3.10. Subiendo el archivo de emisor a la placa del micro-controlador

Se presiona el icono que dice “Subir” y se ejecutaran 2 acciones al mismo tiempo que son la compilación del programa y su carga a la placa, el proceso de las dos tareas se observara en la parte inferior de la ventana y lo mostrara visualmente mediantes una barra de color que se tendrá que llenar completamente.



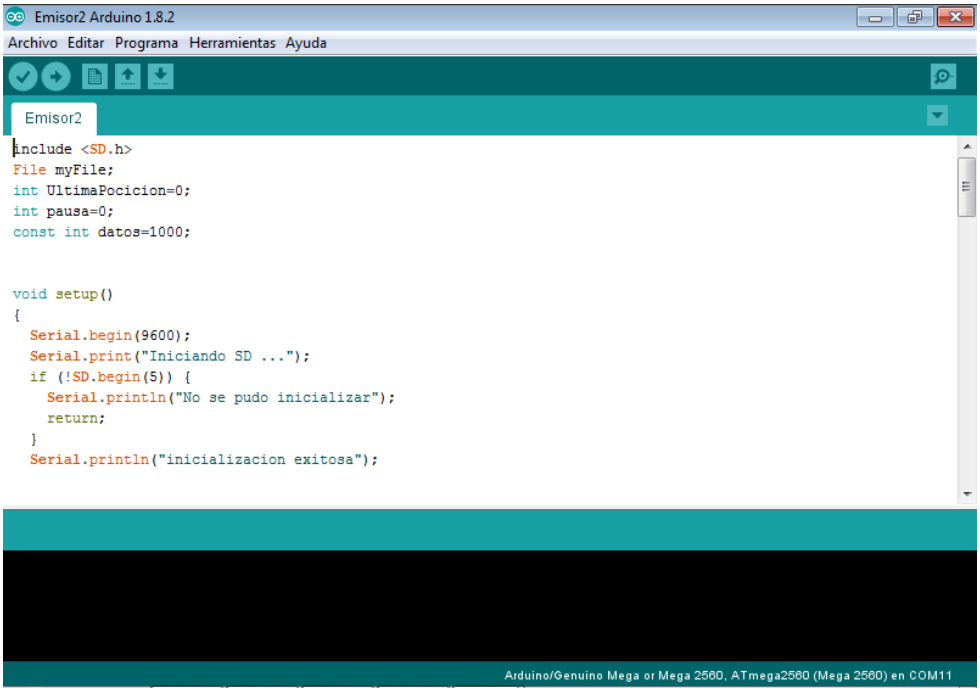
3.11. Mensaje de archivo cargado con éxito

Si el programa no tiene ningún error y el código se sube correctamente nos lo mostrara al final de la pantalla con un mensaje de “Subido”, si hay un error en color de ese recuadro se tornara en rojo.

Los códigos correspondientes de la parte del sistema emisor y receptor son los siguientes:

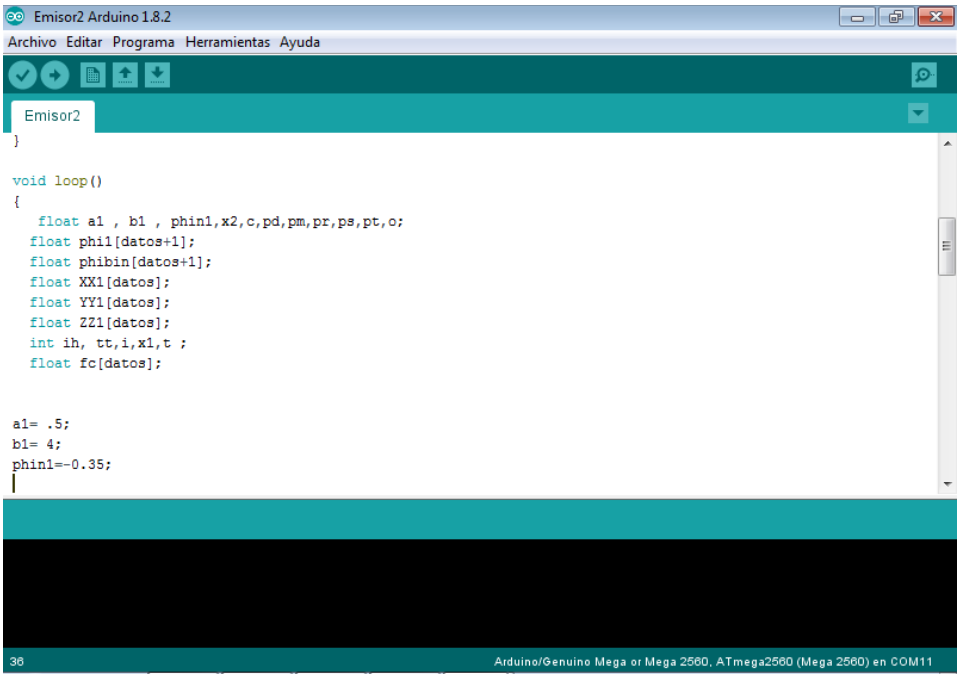
3.4.1.- Código emisor

En esta parte podemos observar el código en la parte de configuración del sistema donde se declaran librerías a utilizar, algunas constantes y se determina la velocidad del puerto serial para trabajar de manera correcta sin que se pierda ningún carácter. Además de esto se incluye un mensaje en caso de que el sistema inicie de manera correcta y que después de este paso proseguirá a continuar con la demás del código, cabe señalar que si esta parte no se ejecuta el código se cicla y no permitirá avanzar de ninguna manera a la siguiente instrucción.



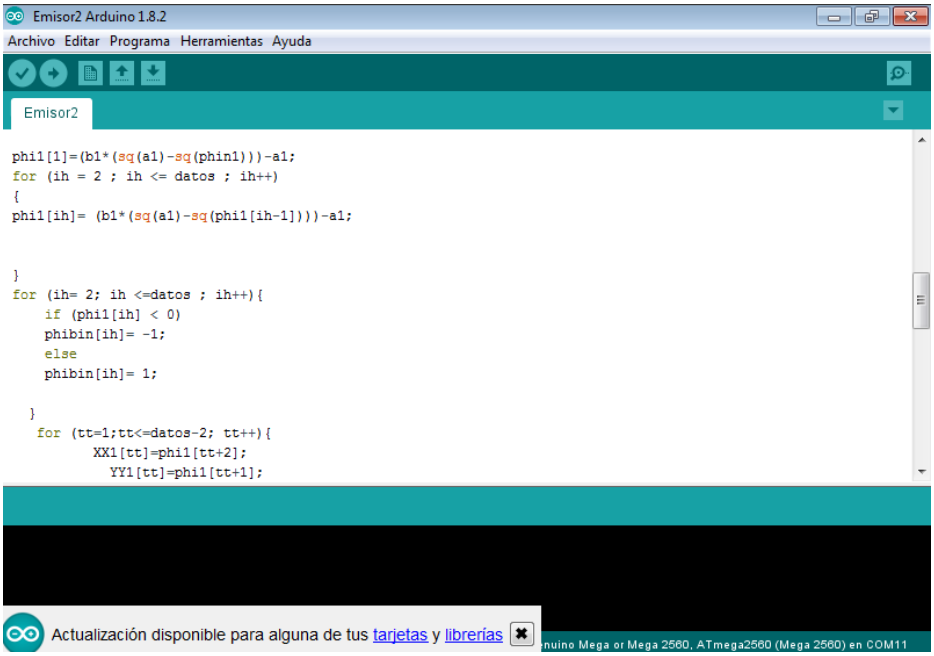
3.12. Parte del código del sistema emisor

Ahora en lo que se puede observar de esta parte del código tenemos el ciclo que se repetirá con el tiempo indefinidas veces hasta que se retire la alimentación del sistema. Tenemos también la declaración de vectores a utilizar y variables así como unas constantes a utilizar en la ecuación principal.



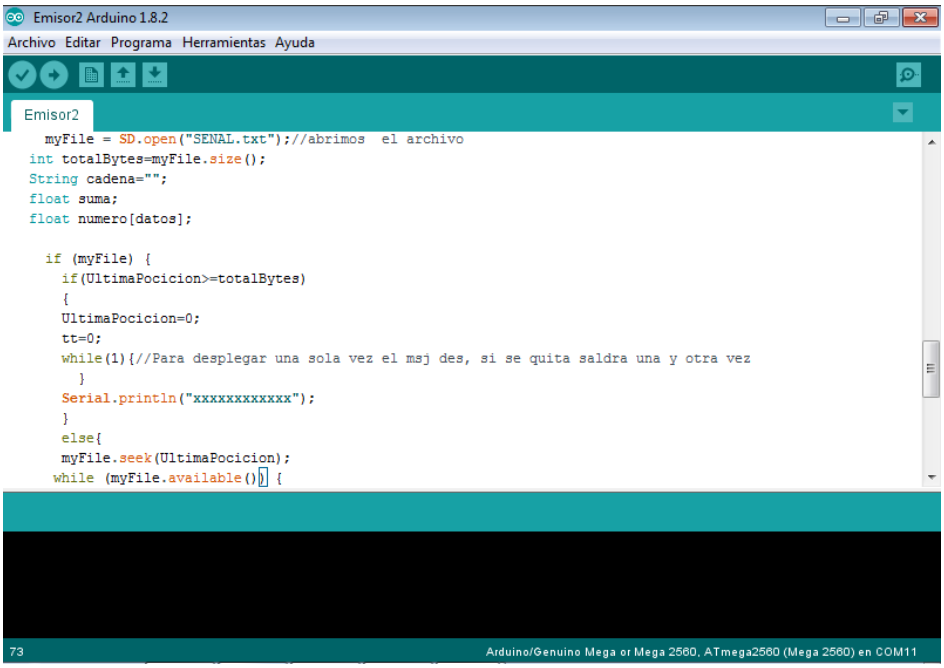
3.13. Parte del código del sistema emisor

Aquí se define la ecuación principal y se procede a iniciar con los ciclos correspondientes para que se empiece a generar el mapa logístico



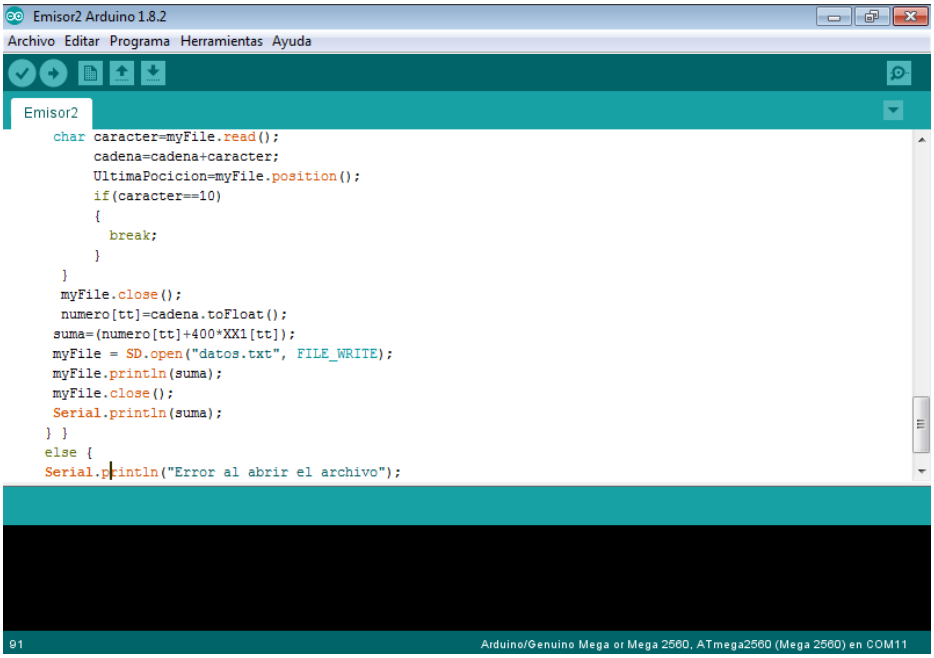
3.14. Parte del código del sistema emisor

En esta parte se observa una instrucción especial que corresponde al módulo SD donde se busca un archivo que exista en la memoria para extraer los datos que contenga y se guarda cada valor en una variable llamada “cadena”. Además se incluye una pequeña parte del código para que el sistema solo determine la longitud del mensaje en la memoria y realice el número de ciclos correspondientes



3.15. Parte del código del sistema emisor

En esta parte el carácter que se extrajo de la memoria proviene en formato “String” y para procesarlo es necesario convertirlo en nuestro caso a “Float”, para ello se realiza la instrucción “String.toFloat”. Otra parte que se observa es donde se genera el criptograma y se envía de nuevo a la memoria SD un archivo con extensión .txt con el nombre de “Datos”



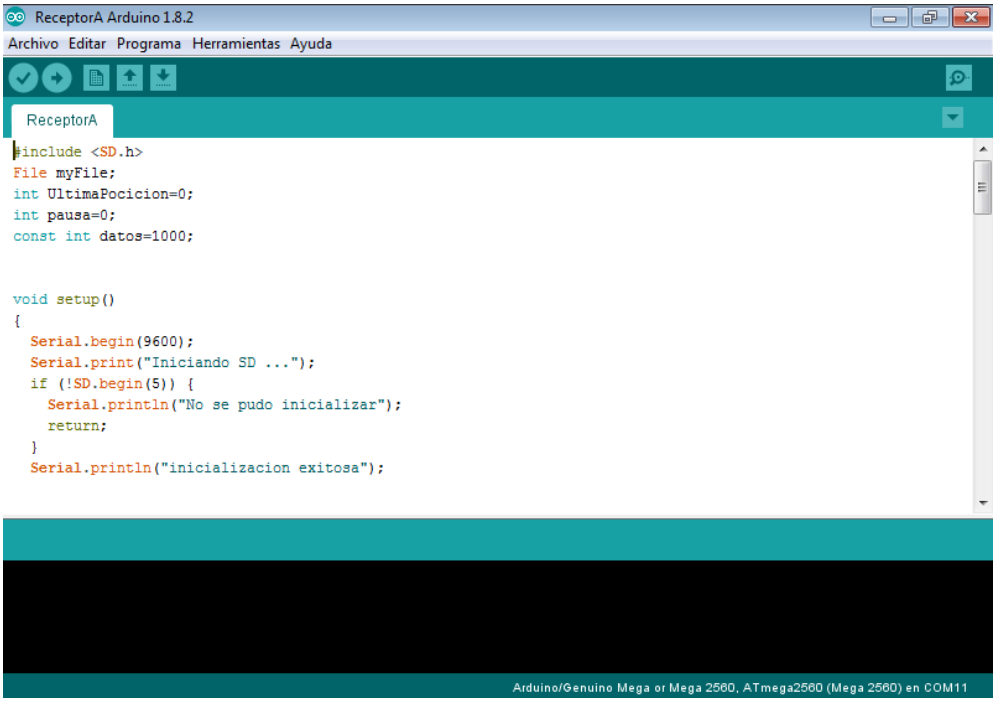
3.16. Parte del código del sistema emisor

3.4.2.- Código receptor

Y por la parte del recetor es la siguiente:

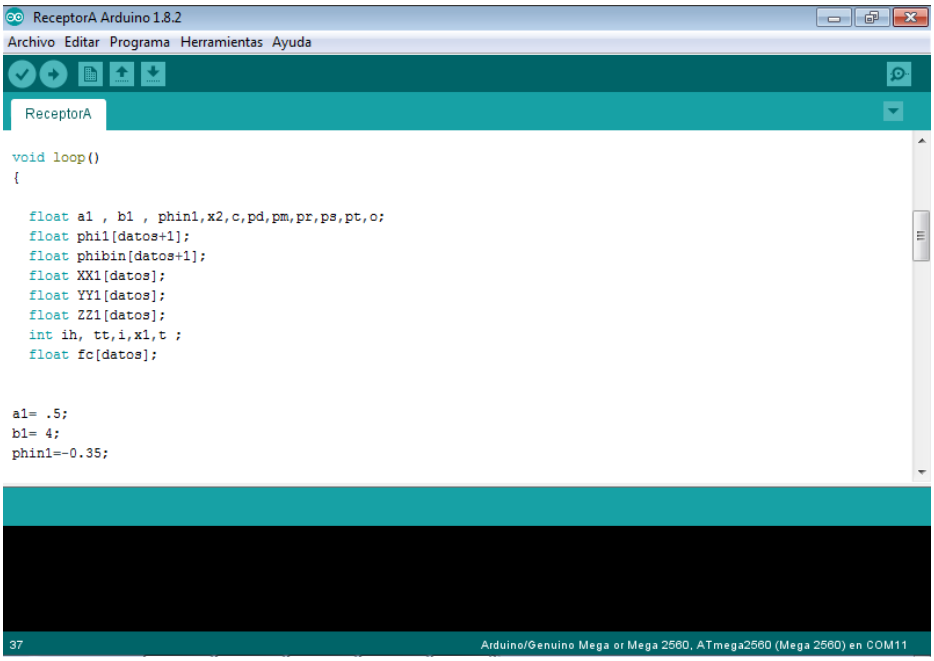
En la parte del receptor también se declaran algunas constantes y las librerías a utilizar en la parte de la configuración y se define por igual la velocidad de comunicación del micro-controlador y el modulo

NOTA: Las velocidades de transmisión de datos tanto en el emisor como en el receptor deberán de ser iguales ya que si se modifica y coloca otra al momento de transmitir los datos se pueden perder algunos caracteres los cuales podrían causar un mal entendimiento de los datos



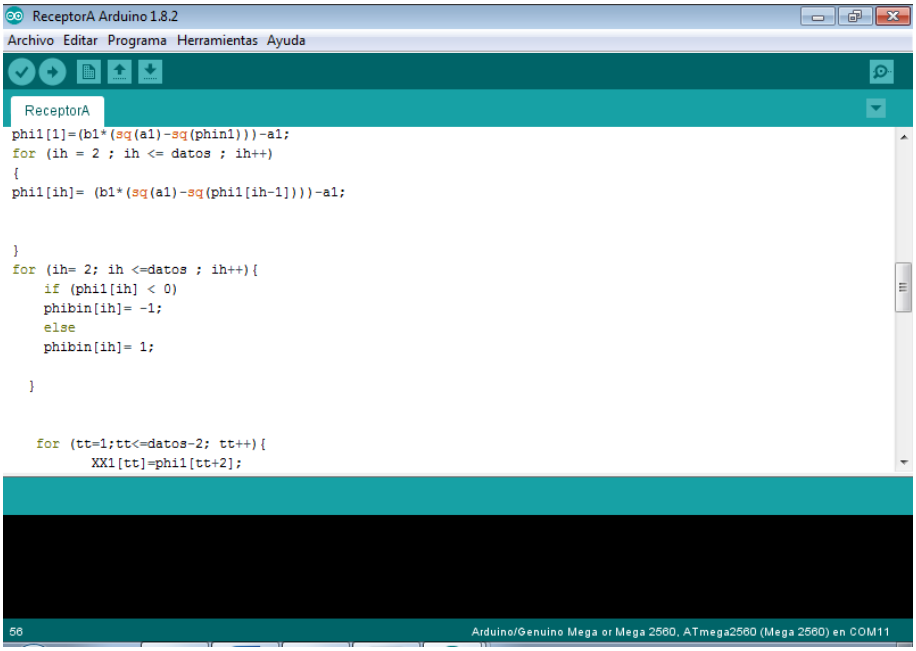
3.17. Parte del código del sistema receptor

Aquí al igual que en la parte del emisor se declaran variables, vectores y constantes a utilizar en las partes posteriores del código



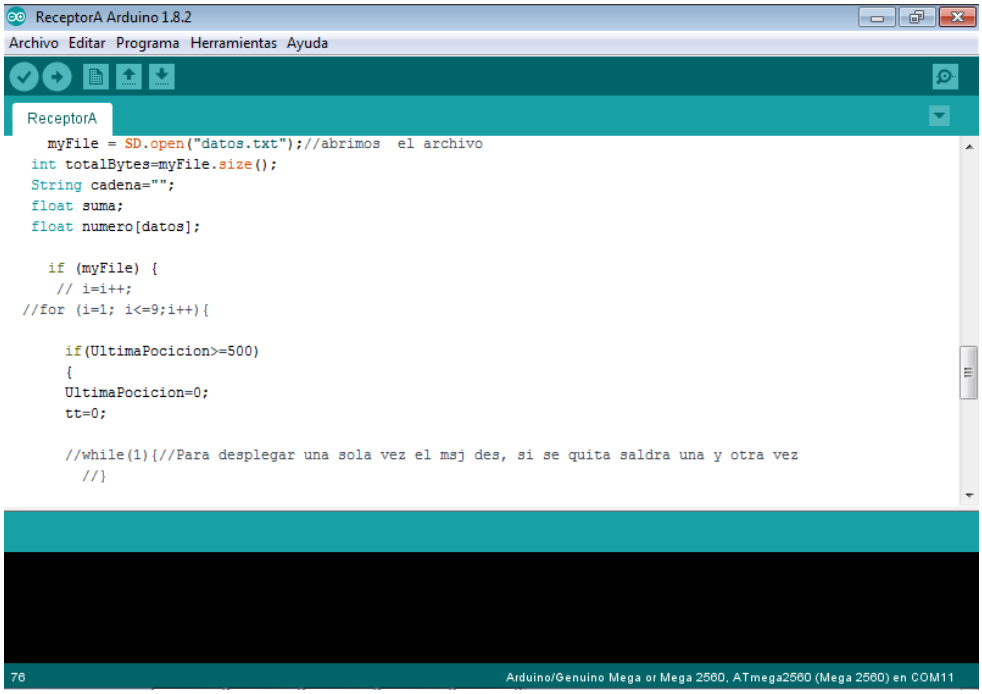
3.18. Parte del código del sistema receptor

Aquí se genera la ecuación del mapa logístico y se inician los ciclos correspondientes para completar las operaciones



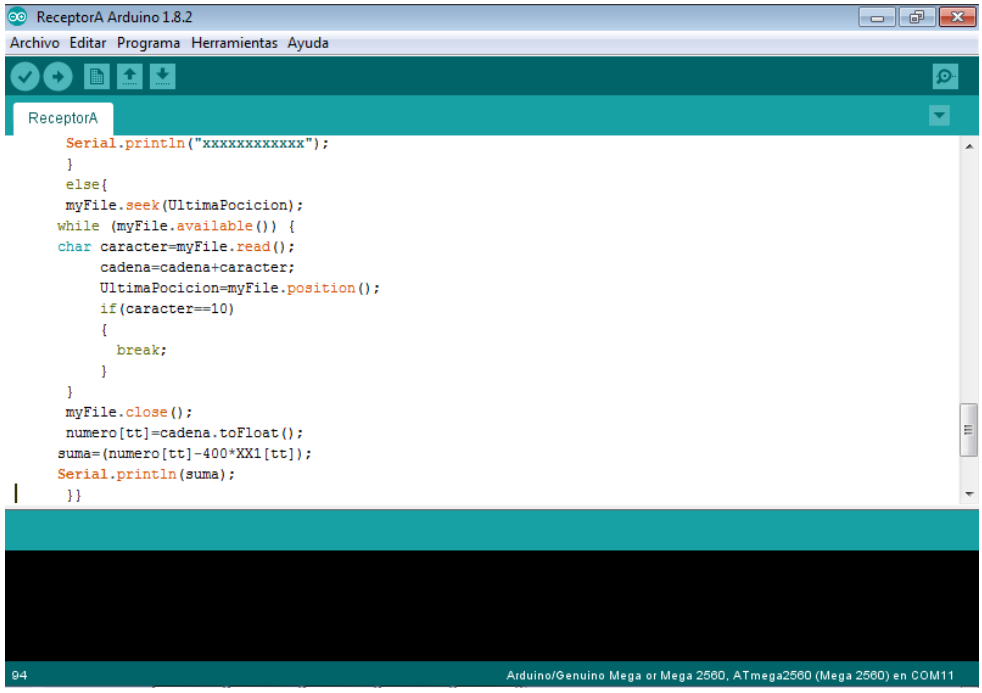
3.19. Parte del código del sistema receptor

Aquí en esta parte se leen los datos que tiene el archivo llamado “Datos” y se guardan en la variable llamada “cadena” que es del tipo “Stirling”, al igual que en el emisor se tiene que cambiar el formato a “Float” para poder trabajar con esta.



3.20. Parte del código del sistema receptor

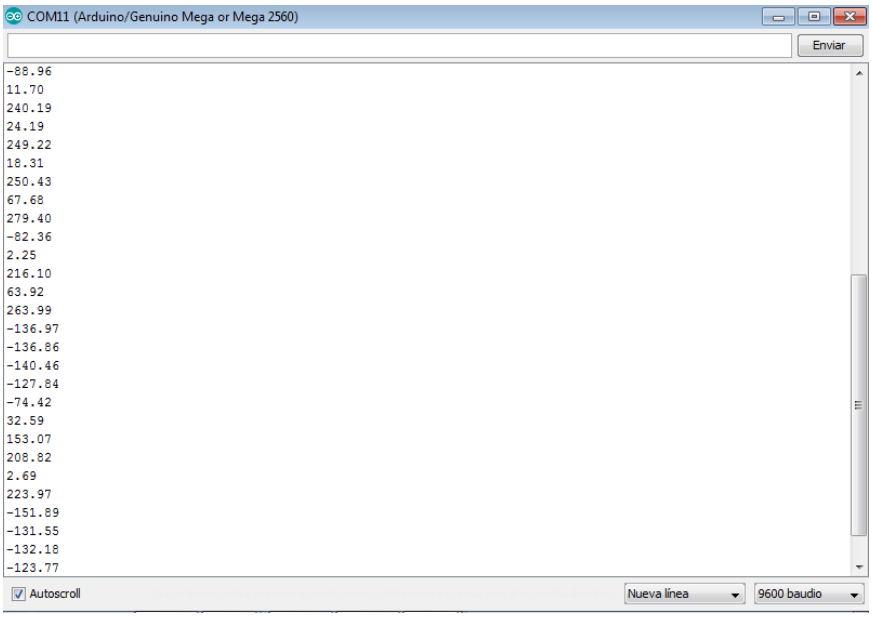
En esta última parte del código lo que se hace es desencriptar el mensaje y desplegarlo tanto en el monitor serial como en el graficador serial para compararlo con el original



3.21. Parte del código del sistema receptor

3.5.- Sistema emisor

Ahora para observar los datos que se están cargando y enviando a la memoria se puede abrir el monitor serial o el graficador serial, en donde se podrán observar los datos de manera numérica o graficados respectivamente. Para abrir el monitor serial se selecciona la pestana de herramientas y después monitor serie. Se debería de observar algo como lo siguiente:

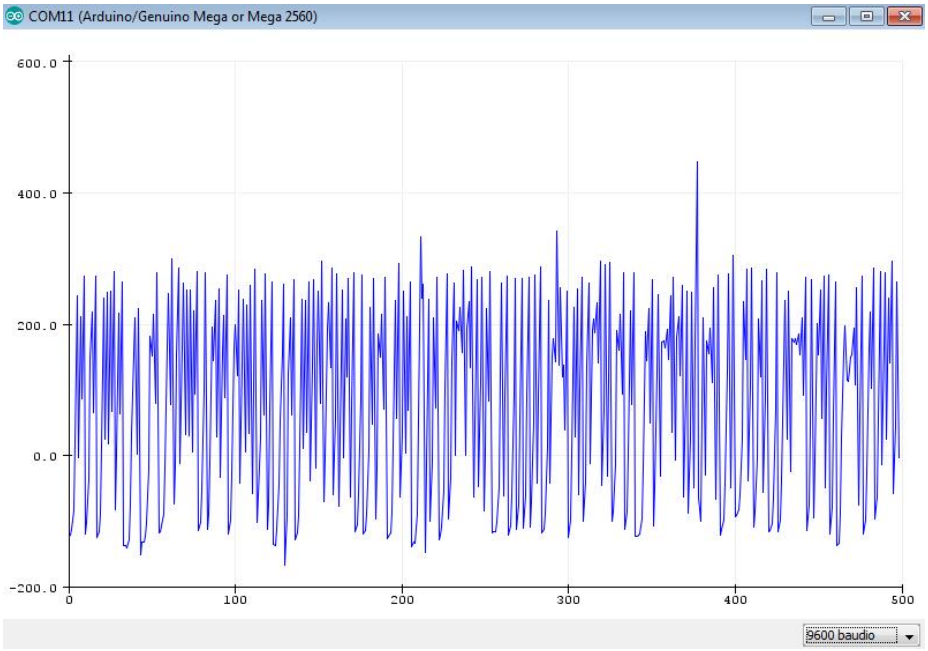


3.22. Valores encriptados del sistema emisor desplegados por el monitor serie

Los datos vistos anteriormente pueden parecer no tener sentido pero esos datos son enviados a la memoria y forman parte del criptograma, debido a que los mismo ya están sumados al sistema caótico y serán los que podrán circular de manera publica por la via de Internet.

Ahora si se desea ver los datos graficados es decir el criptograma en la misma pestana se presiona ahora el Graficador Serie,

NOTA: No se podrá abrir el monitor serial y el graficador serial al mismo tiempo, si se realiza esa acción el programa le dara un aviso y lo que deberá de realizar será primero cerrar uno si desea abrir el otro.



3.23. Valores encriptados y graficados desplegados por el graficador serial

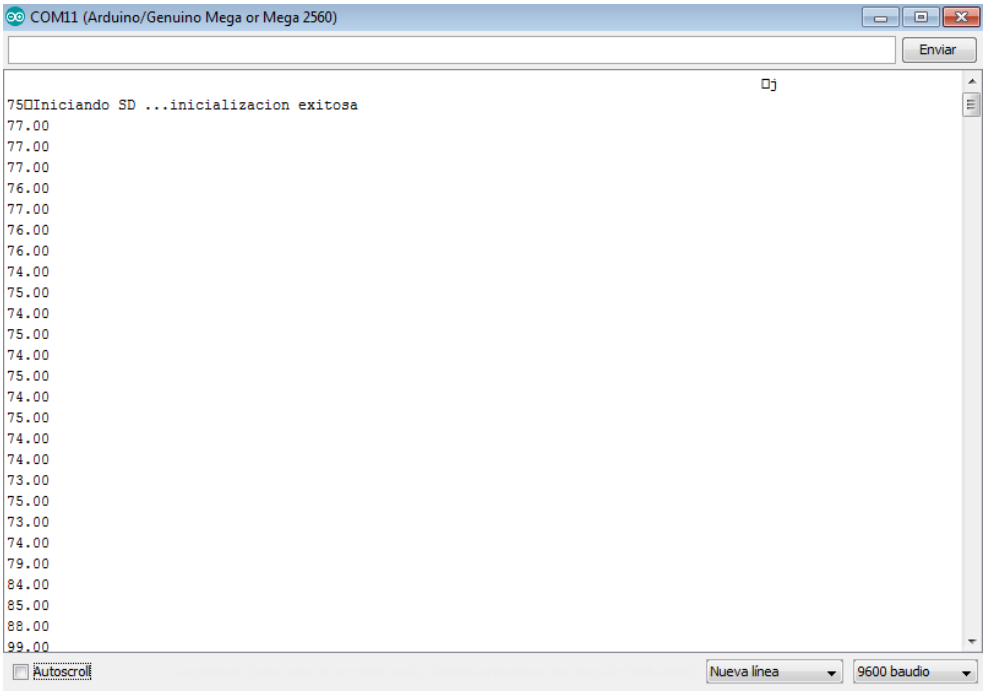
Esta imagen muestra de manera grafica el criptograma que son los datos enviados a la SD y que son los que se enviaran por el canal publico.

Una vez que el programa se ejecuto y debido a limitantes descritas en capitulo anterior solo se desplegaran 950 datos y no realizara otra acción.

3.6.- Sistema receptor

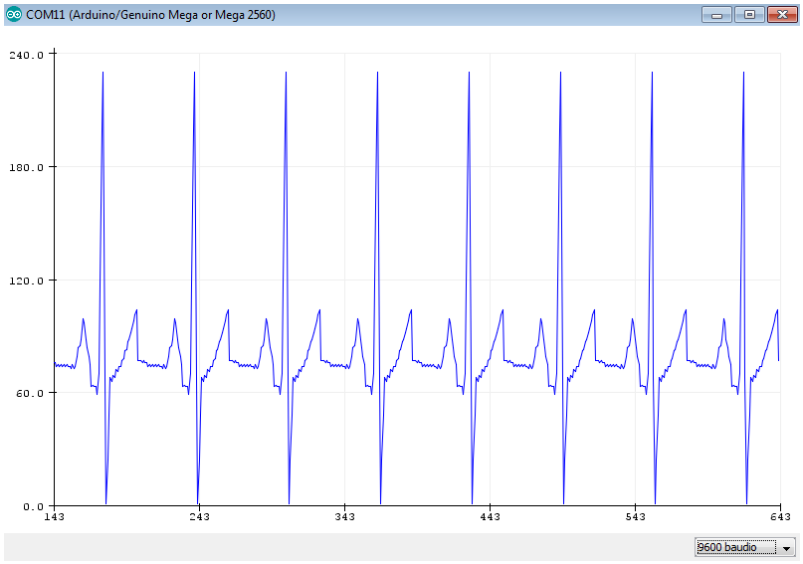
Ahora veremos la parte del receptor.

El proceso de abrir el archivo, compilar y subir el código será el mismo que el del emisor solo que esta vez se buscara el archivo de ReceptorA, después de abrir, compilar y cargar se abre el monitor para observar los datos que se reciben, eso se realiza de la misma manera que en el emisor y se tendría que observar lo siguiente:



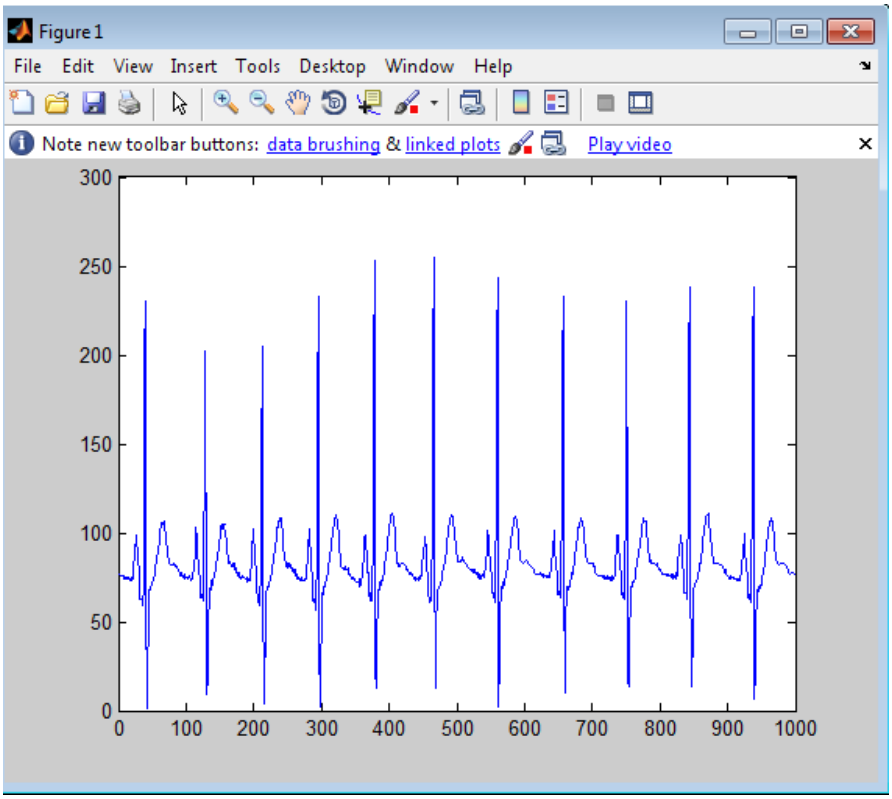
3.24. Valores des encriptados mostrados en el monitor serial del receptor

Lo que se observa son los datos que envía el usuario que esta en la parte del emisor, la manera en que podríamos saber si están correctos en ahora observarlos por medio del graficador serial.



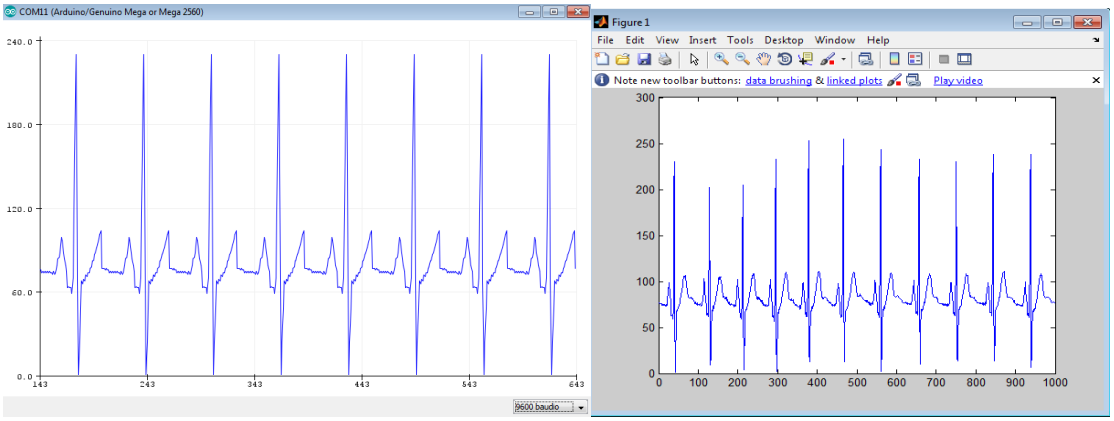
3.25. Mensaje recuperado en el sistema receptor

Podemos observar que son pulsos de una señal de Electrocardiograma y podemos corroborar que están correctos si graficamos el archivo original en el Software de MatLab, asi como lo siguiente:



3.26. Mensaje visto en MatLab

Si se observa hay unas pequeñas diferencias pero son debido a que las operaciones que realiza el Software Arduino las redondea a solo 2 decimales lo que provoca unos pequenos cambios a las graficas, pero que son apenas visibles.



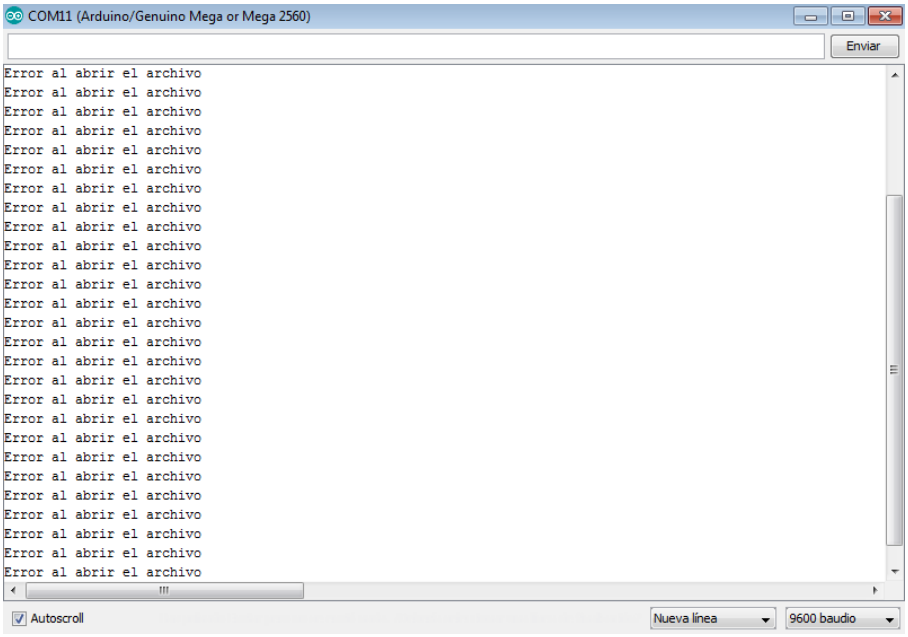
3.27 Comparación mensaje recuperado en sistema criptográfico y mensaje visto en MatLab

3.7.- Errores comunes y soluciones posibles

NOTA:

Hay algunos errores que se desplegaran y que se deberán de tomar en cuenta para poder determinar por que se provocan y posteriormente corregirlos. Los errores mas comunes son los siguientes:

Error al abrir el archivo



3.28 Error del sistema al abrir el archivo

Este error se puede provocar por varias razones

a) Falso contacto o un error de conexión en el puerto de comunicación

Posible solución:

- Revisar las diferentes conexiones del modulo SD como por ejemplo
  - Los pines MISO, MOSI, SCK, CS

b) Falta de voltaje de alimentación o de tierra

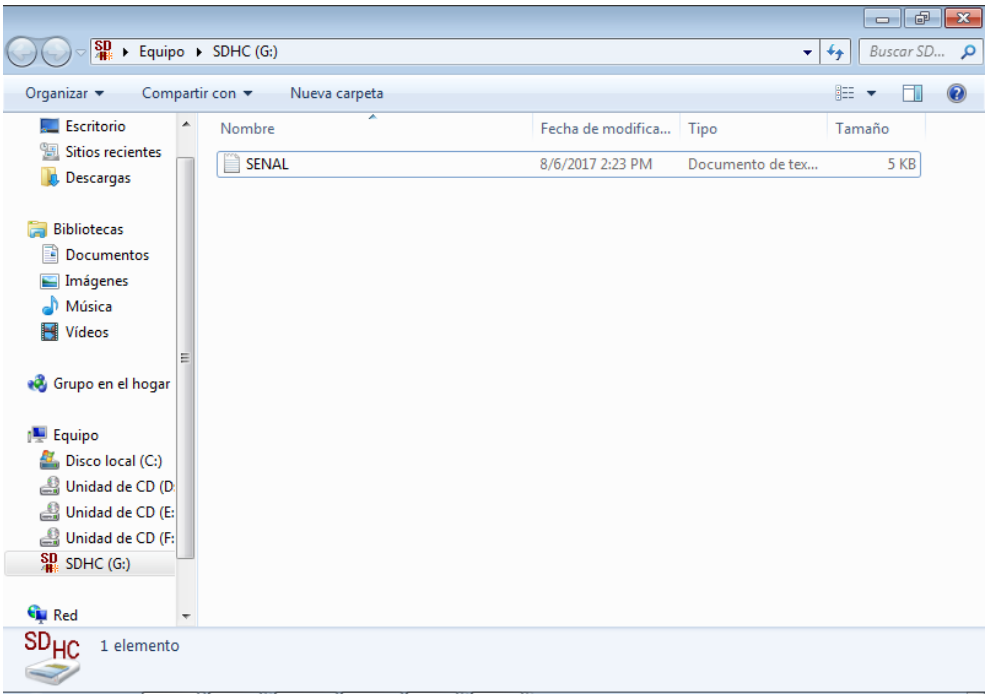
Posible solución:

- Verificar la continuidad de los cables o el falso contacto de uno de estos pines

c) El archivo con los datos del mensaje no esta agregado en la memoria SD

Posible solución:

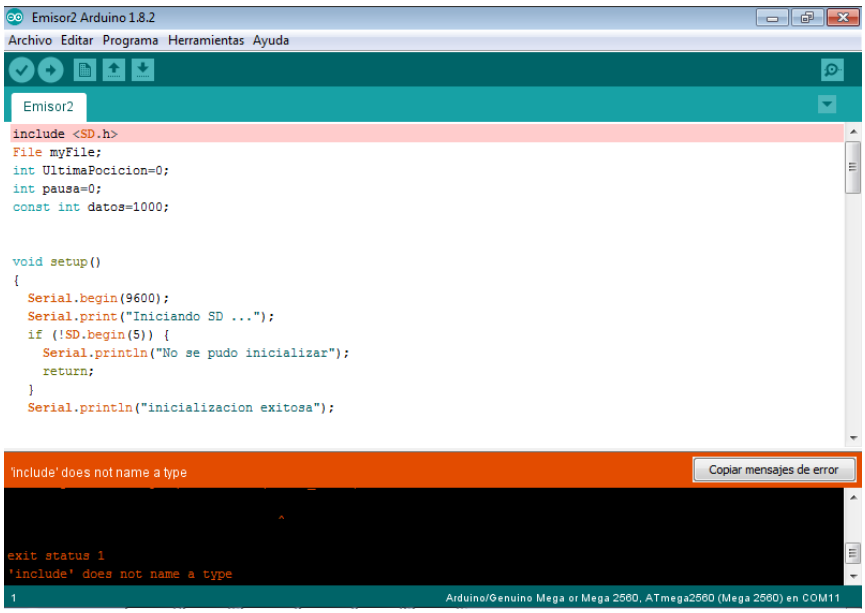
- Revisar que el archivo con el nombre SENAL y extensión .txt este colocado en la memoria para que el sistema pueda extraer los datos del mismo



3.29 Revisión de archivo “SENAL” en memoria SD

- Revisar que exista el espacio suficiente para colocar el programa en la memoria y además para poder agregar los datos del criptograma

Error al compilar el programa:



3.30. Error al compilar código en Software Arduino

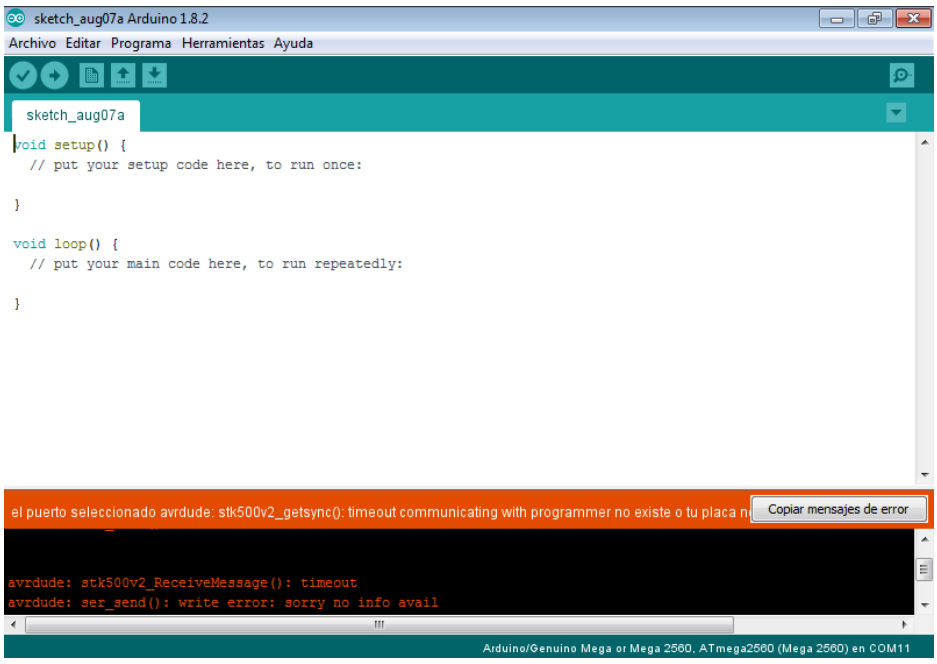
Este error se provoca por:

a) Se realizo una modificacion alguna parte del código y existe algún error en la escritura de una acción.

Posible solución:

- Revisar cada detalle del código para corroborar de que no se esta cometiendo un error al escribir un comando o verificar el cambio de nombre de alguna variable al no modificarse en la parte de configuración y viceversa
- Revisar que las librerías que se deben utilizar en el código estén agregadas al inicio del programa.

Error al subir el programa a la placa:



3.31. Error al subir código a la placa de pruebas

Este error se puede provocar por:

a) La placa del micro-controlador no esta conectada a la computadora

Posible solución:

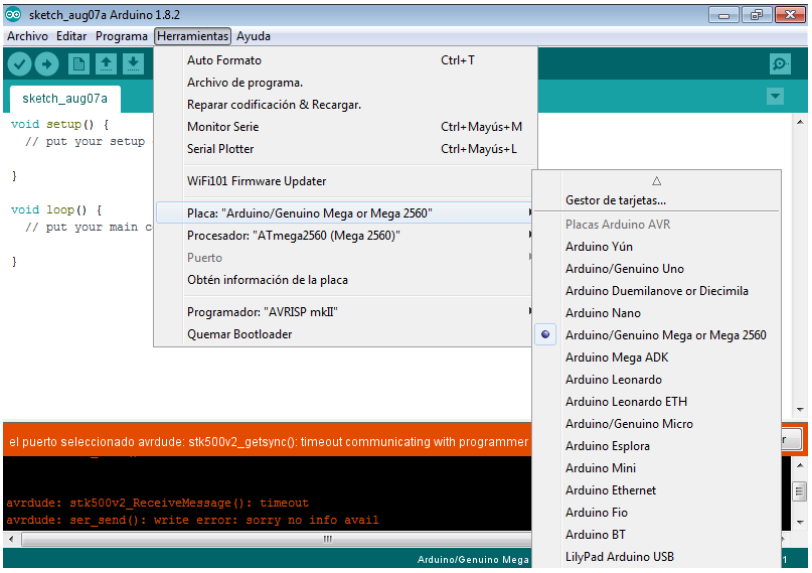
- Verificar la conexión física de la placa del micro-controlador con la computadora

b) El cable de transmisión de datos esta danado.

Posible solución:

- Usar otro cable para cargar el programa y corroborar que no este danado por dentro.

c) No esta seleccionada la placa correctamente en el menú de herramientas



3.32. Selección de placa para subir código a la placa

Posible solución:

- Para corregir este error se deberá ingresar al menú de herramientas, después se selecciona la “Placa” correcta, esto dependerá de la placa con la que se trabaje y además después de seleccionar la placa se deberá se escoger el procesador correcto que también dependerá de la placa que se utilice

d) Existe algún corto o puente de pines en la placa

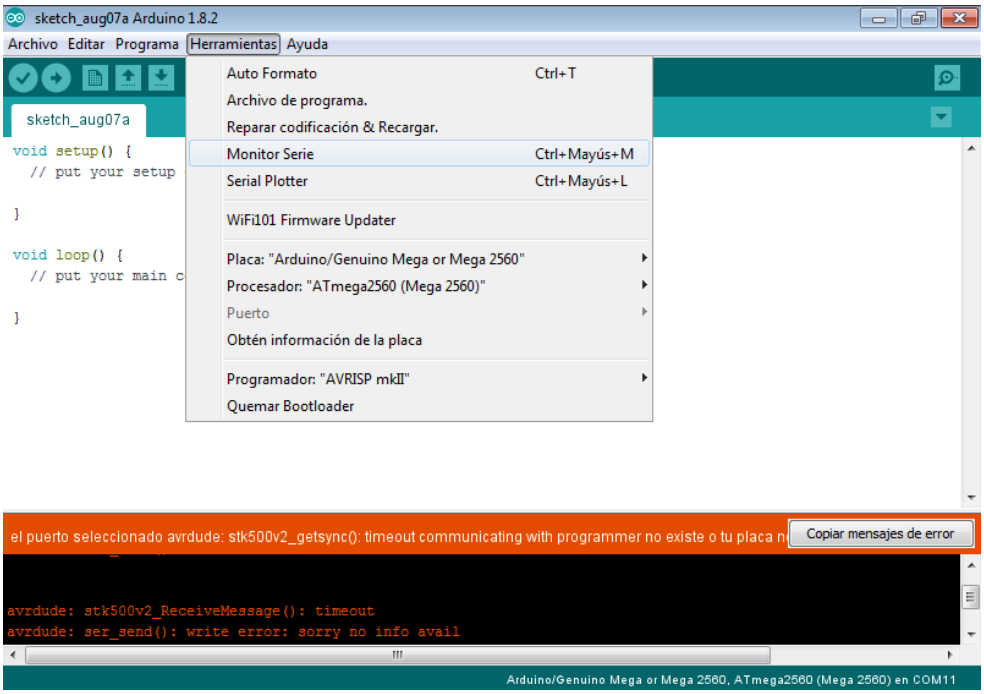
Posible solución:

- Hacer una revisión visual de todo el circuito, verificar que no exista algún puente o cable a tierra y si no se observa con la simple vista se puede ir desconectando cable por cable y cargando a la misma vez el programa para saber que cable es el que genera el problema

e) Se conectó o se usó algún puerto de comunicación que esta destinado a usarse por el micro-controlador internamente, esto provoca que al querer ingresar datos por un puerto que se encuentra en comunicación, se interfiere y falla

Posible solución:

- Verificar que en la placa del micro-controlador y código no se este usando un puerto de comunicación interna tal como los Rx y Tx, si es el caso desconectarlos y volver a intentar



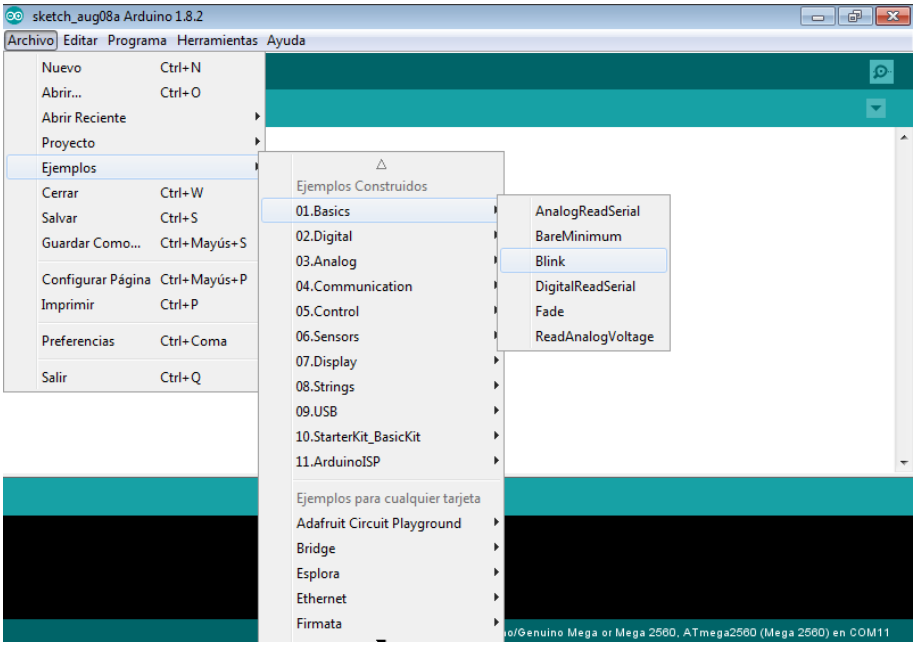
3.33. Error al abrir monitor serie

f) Esta abierto alguna de las siguientes ventanas:

- Monitor Serial
- Graficador serial

Posible solución:

- Cerrar las ventanas de monitor serial o graficador serial e intentar de nuevo cargar el código a la placa del micro-controlador
- Cerrar y abrir de nuevo todo el Software de Arduino para verificar que no sea un error del mismo sistema de computo.
- Intentar cargar uno de los programas mas sencillo que vienen como ejemplos del Software Arduino tal como el de “Blink”, es un código sencillo que si se tiene problema al cargar se deberá de realizar el paso 2 de este error



3.34. Prueba de carga de archivo básico en la placa

Capitulo IV

4.1.- Conclusiones:

El sistema de encriptamiento basado en el Micro-controlador Arduino cumplió con el requisito de poder adquirir una señal de entrada y catalogarla como el mensaje, a su vez el sistema genera el mapa logístico el cual procesa junto con el mensaje para poder generar el criptograma y posteriormente enviarlo vía internet o públicamente. Esto es la parte del emisor, por su parte el receptor tiene un sistema caótico que hace el proceso inverso al emisor para poder recuperar el mensaje enviado. El sistema de micro-controlador es básico y cumple con las expectativas de su desempeño total.

4.2.- Trabajo a futuro:

Como trabajo a futuro se podría tener en cuenta exportar el código a un micro-procesador para poder adquirir una mayor potencia, velocidad y rendimiento del mismo, además de que la comunicación con el correo electrónico sea por ambas partes, tanto del emisor como del receptor para que el sistema posea una mayor facilidad de manejo sea más amigable la comunicación Usuario-Maquina

4.3 - Bibliografía:

[1].- Pablo Cazau, 2002-10-09, extraido de URL: http://antroposmoderno.com/antro-articulo.php?id_articulo=152

[2].- Extraido de URL: https://www.uco.es/dptos/quimica-fisica/quimica-fisica/MC/QC2.htm

[3].- Extraido de URL: https://es.wikipedia.org/wiki/Teor%C3%ADa_del_caos

[4]. German Ros Sanchez, extraido de URL: http://webs.um.es/jmz/DiseGrafSimula/alumnos_08_09/german_ros/index.files/fractal1_Intro%201

[5].- Extraido de la pagina principal de Arduino.cl, URL: http://arduino.cl/arduino-mega-2560/

[6].- Extraido de la pagina principal de Arduino.cl
https://www.arduino.cc/en/Reference/SPI

[7].- Extraido de la pagina principal de Arduino.cl
https://www.arduino.cc/en/Reference/SDCardNotes

[8].- http://es.ccm.net/contents/610-fat16-y-fat32

[9].- https://www.luisllamas.es/arduino-puerto-serie/