**GUIA ARDUINO C PARTE1**

**NICOLAS AQUILES DURANGO GOEZ**

* P1: ¿[Qué tamaño tiene el tipo de dato double](https://www.arduino.cc/en/Reference/HomePage)?

Doble precisión flotante número de punto. En el Uno y otras placas base ATMEGA, este ocupa 4 bytes.

Por Arduino Due, dobles tienen de 8 bytes (64 bits) de precisión.

Compare el tipo double con el tipo float, ¿Qué puede concluir?

En el Arduino ambos tipos de datos representan las misma cosa ocupan el mismo espacio

En el Arduino, doble es del mismo tamaño que flotador.

* P2: Explore algunas de las [bibliotecas](https://www.arduino.cc/en/Reference/Libraries) estándar de arduino. Seleccione tres de ellas y explique para qué sirven y de un ejemplo cada una.

**WiFi biblioteca**

Con el Arduino WiFi Shield, esta biblioteca permite que una placa Arduino se conecte a internet.

Puede servir como un servidor de aceptar conexiones entrantes o un cliente que realiza los salientes.

La biblioteca es compatible con WEP yWPA2 cifrado personal, pero no WPA2 Enterprise. También tenga en cuenta, si el SSID no se transmite, el escudo no se puede conectar.

El WiFi biblioteca es muy similar a la [de Ethernet](https://www.arduino.cc/en/Reference/Ethernet) biblioteca, y muchas de las llamadas a funciones son las mismas.

Ejemplo

[ScanNetworks](https://www.arduino.cc/en/Tutorial/ScanNetworks): Muestra todas WiFi redes en rango

### **Conectado con WPA**

En este ejemplo se analiza en busca de redes 802.11b / g con el Arduino WiFi escudo. Monitor de serie de la Arduino imprimirá información sobre el tablero y las redes se pueden ver.

No va a conectarse a una red.

## Biblioteca de GSM

* [GSM](https://www.arduino.cc/en/Reference/GSM) para la conexión a una red / GRPS GSM con el escudo GSM.

Permite a una placa Arduino para hacer la mayoría de las operaciones que se pueden hacer con un teléfono GSM: realizar y recibir llamadas de voz, enviar y recibir SMS, y conectarse a Internet a través de una red GPRS.

Ejemplo

Hacer una llamada de voz: obtener su escudo para hacer llamadas telefónicas desde el Serial Monitor

## Biblioteca SD

* [SD](https://www.arduino.cc/en/Reference/SD) - para la lectura y escritura de tarjetas SD

Permite la lectura y escritura en tarjetas SD, por ejemplo, en el Escudo de Ethernet de Arduino

La biblioteca soporta FAT16 y FAT32sistemas de archivos en tarjetas SD estándar y tarjetas SDHC

Los nombres de los archivos que se pasan a las funciones de la librería SD pueden incluir caminos separados por forward-barras, /, por ejemplo, "directorio / filename.txt".

ejemplo

### **Utilización de la biblioteca SD para recuperar información sobre un puerto serie**

Este ejemplo muestra cómo leer un archivo de una tarjeta SD usando la biblioteca SD y enviarlo a través del puerto serie.

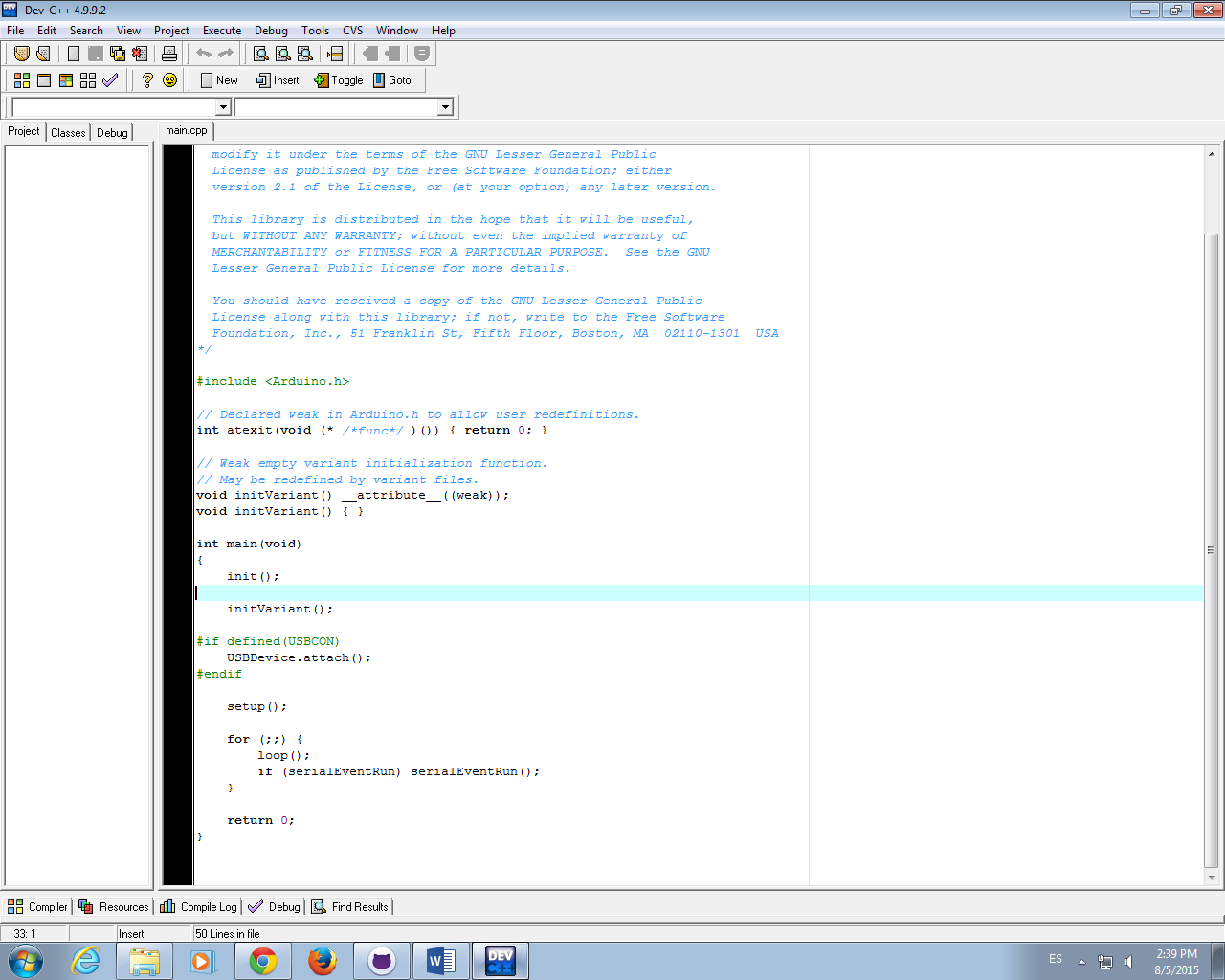
* P3: ¿Cuál es la [estructura](https://www.arduino.cc/en/Reference/HomePage) de un programa para un Arduino?
* [setup](https://www.arduino.cc/en/Reference/Setup)()
* [loop](https://www.arduino.cc/en/Reference/Loop)()
* P4: ¿Cuál es la función de cada una de las partes de la estructura de un programa en Arduino?

La función setup () se llama cuando se inicia un boceto, se utiliza para inicializar variables, modos de pasador,

Una función setup (),inicializa y establece los valores iniciales.

La función loop () hace exactamente lo que su nombre indica (lazo), y los bucles de forma consecutiva, lo que permite el programa para cambiar y responder. Lo utilizan para controlar activamente la placa Arduino

* 5: localice el archivo main.cpp en el directorio donde descomprimió el archivo .zip descargado de la página del proyecto Arduino. No dude en revisar cada directorio. La función main en un programa en C es el punto de entrada del programa.



P5a. ¿Cuántas veces se ejecuta la función setup?

Se ejecuta una vez

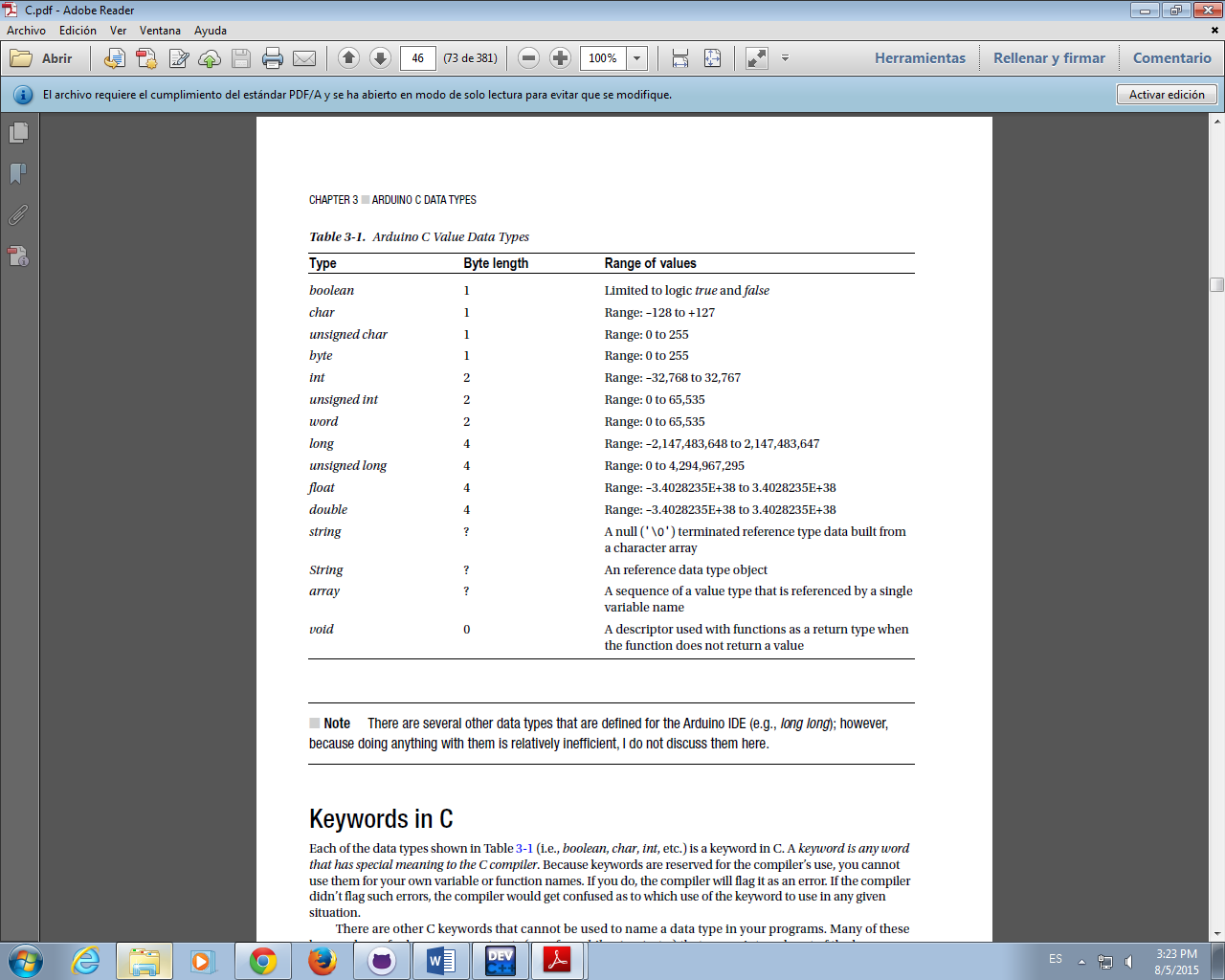
P5b. ¿Cuántas veces se ejecuta la función loop?

Se ejecuta cuantas veces se necesite indefinidamente.

P5c. ¿Cuál es la ruta del archivo main.cpp en el sistema de archivos?

C:\Users\Administrador\Downloads\arduino-1.6.5-r2-windows\arduino-1.6.5-r2\hardware\arduino\avr\cores\arduino

P6. ¿Qué tamaño en bytes y qué rango de valores tiene los siguientes [tipos de datos](https://www.arduino.cc/en/Reference/HomePage): boolean, char, unsigned char, byte, int, unsigned int, word, long, unsigned long, float, double, string, String, array, void.



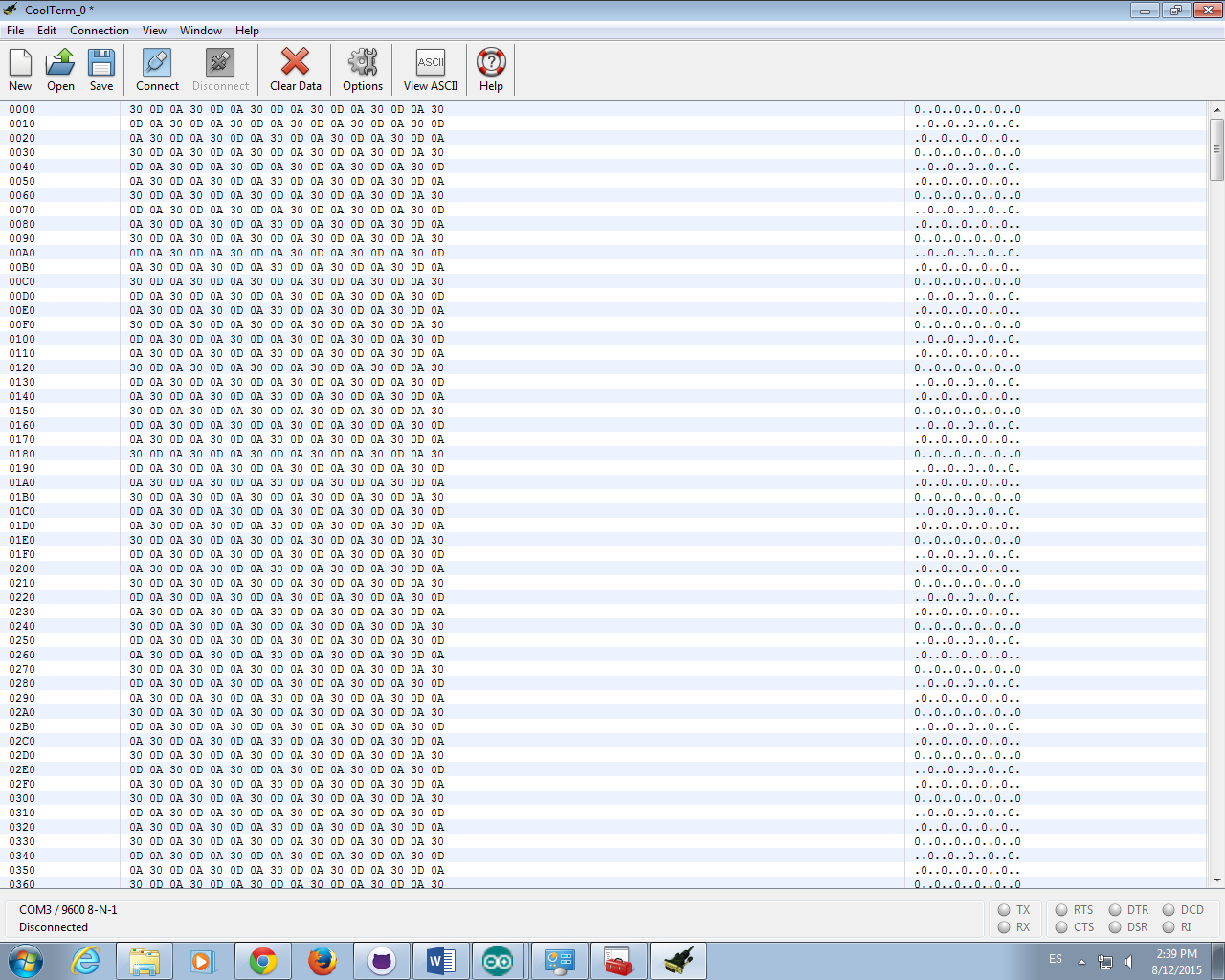
P7. ¿Para qué sirve terminar un [string](https://www.arduino.cc/en/Reference/String) en null (código [ASCII 0](http://www.asciitable.com/))?

En general, las cadenas se terminan con un carácter nulo (código ASCII 0). Esto permite que las funciones (como Serial.print ()) para saber dónde el final de una cadena es. De lo contrario, podrían seguir leyendo bytes posteriores de la memoria que no son en realidad parte de la cadena.

P8. ¿Cuál es el código ASCII para los número del 0 al 9?

0x30 hasta 0x39

P10. Explique la diferencia entre ambos. Utilice como referencia una tabla de códigos [ASCII](http://www.asciitable.com/).



Nos muestra igual resultado en cool term que en el monitor arduino

30=0 en hexadecimal

0d= retorno del carro

0A=coloca una nueva línea

Luego el programa vuelve y repite el 0 en hexadecimal

P11. ¿Cómo se crean un arreglo? muestre y explique un ejemplo.

int myInts [6];

int myPins [] = {2, 4, 8, 3, 6};

int mySensVals [6] = {2, 4, -8, 3, 2};

mensaje char [6] = "hola";

Puede declarar una matriz sin inicializarla como en myInts.

Con Int declaramos un arreglo, existen varios métodos según lo que necesitemos, ejemplo en myInts declaramos una matriz sin inicializarla.

En myPins declaramos un arreglo sin elegir explícitamente un tamaño. El compilador cuenta los elementos y crea una matriz de tamaño adecuado.

Por último, puede tanto inicializar y tamaño de la matriz, como en mySensVals. Tenga en cuenta que cuando se declara una matriz de tipo char, se requiere un elemento más de su inicialización, para mantener el carácter nulo necesario.

P12. ¿Cómo se accede y cómo se escribe un elemento de un arreglo? muestre y explique un ejemplo.

Las matrices se **indexan cero**, es decir, refiriéndose a la inicialización de la matriz anterior, el primer elemento de la matriz está en el índice 0, por lo tanto, mySensVals [0] == 2, mySensVals [1] == 4, y así sucesivamente.

Para asignar un valor a una matriz

mySensVals [0] = 10;

P13. Cree un arreglo con su nombre y escriba un programa que envíe cada uno de los caracteres del nombre al puerto serial. Observe el resultado utilizando CoolTerm.

char myejemplo[7] = {'N', 'I', 'C', 'O', 'L','A','S'};

void setup() {

// put your setup code here, to run once:

Serial.begin(9600);

}

void loop() {

// put your main code here, to run repeatedly:

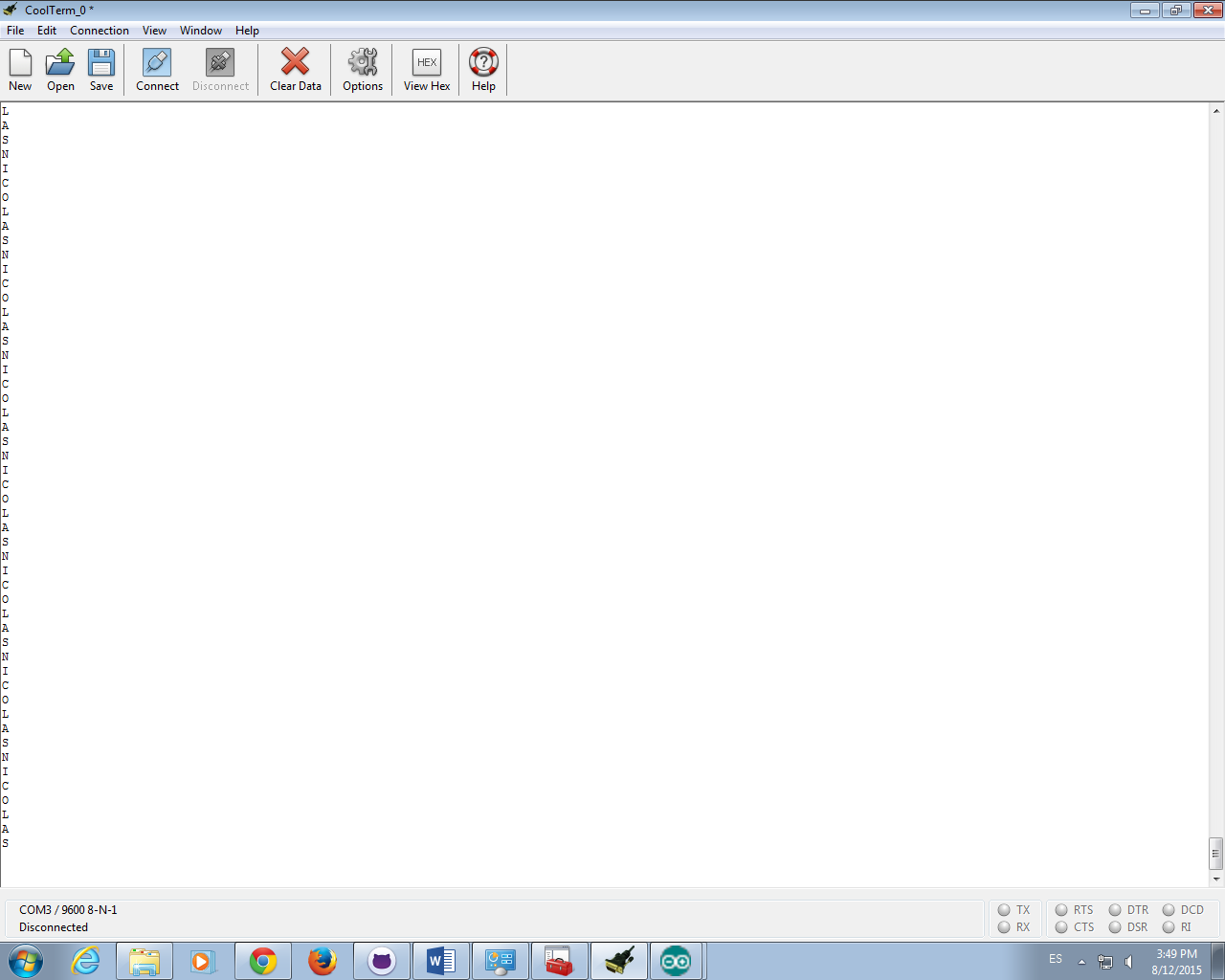
int i;

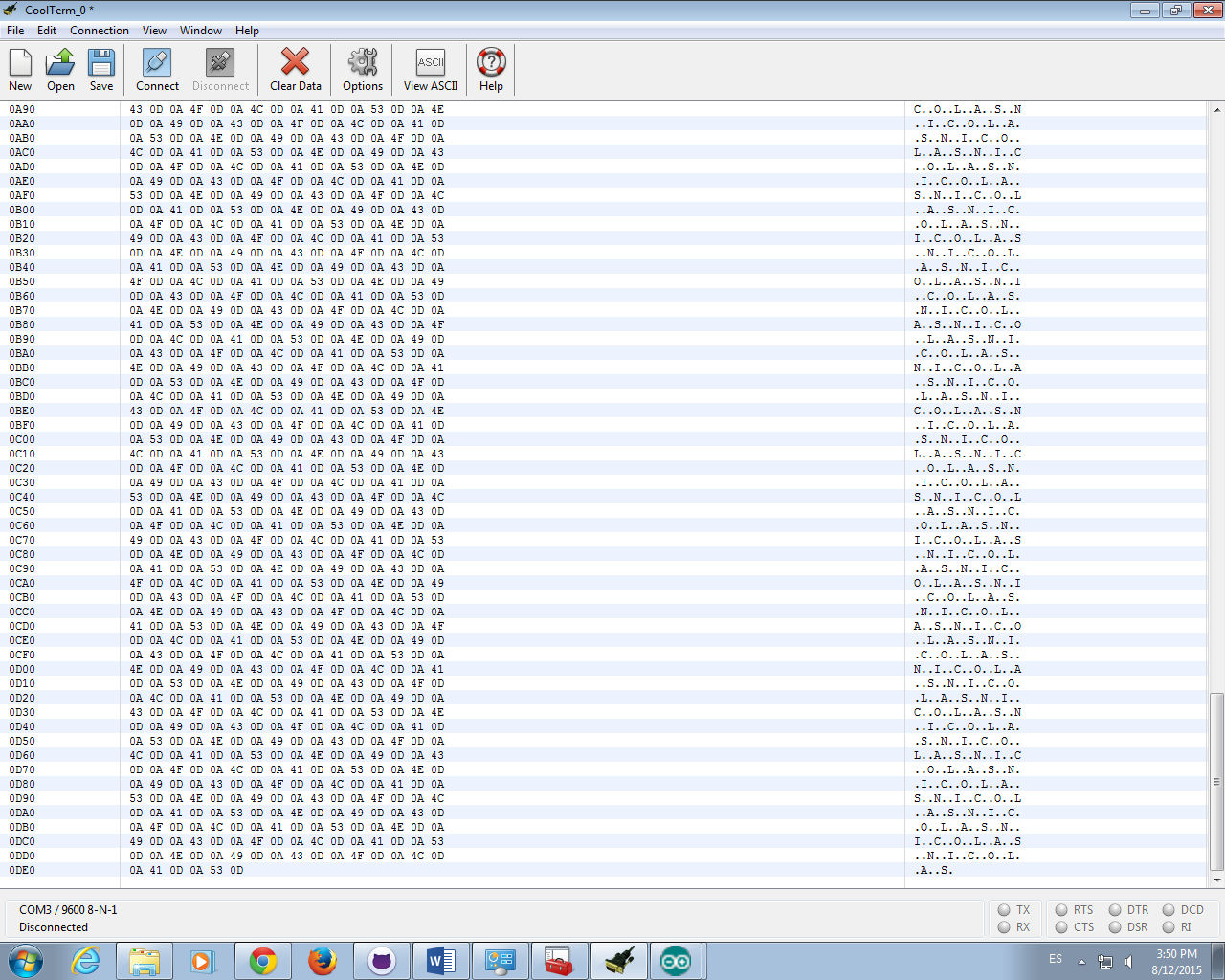
for (i = 0; i < 7; i = i + 1) {

Serial.println(myejemplo[i]);

}

}





Identifique las posibles [estructuras de control](https://www.arduino.cc/en/Reference/HomePage) con los que cuenta para programar.

P14. ¿Cuáles son los operadores de comparación? de ejemplos de cada uno

#### **Control Structures**

[if](https://www.arduino.cc/en/Reference/If)

[if...else](https://www.arduino.cc/en/Reference/Else)

[for](https://www.arduino.cc/en/Reference/For)

[switch case](https://www.arduino.cc/en/Reference/SwitchCase)

[while](https://www.arduino.cc/en/Reference/While)

[do... while](https://www.arduino.cc/en/Reference/DoWhile)

[break](https://www.arduino.cc/en/Reference/Break)

[continue](https://www.arduino.cc/en/Reference/Continue)

[return](https://www.arduino.cc/en/Reference/Return)

[goto](https://www.arduino.cc/en/Reference/Goto)

P15. Construya programas ejemplo para las siguientes estructuras de control: if, if-else, if-else if- else, for, switch-case, while.

PARA IF, IF-ELSE

if (x < 200)

{

// action A

}

else

{

// action B

}

if (x < 500)

{

// do Thing A

}

else if (x >= 1000)

{

// do Thing B

}

else

{

// do Thing C

}

PARA FOR

void loop()

{

for (int i=0; i <= 255; i++){

analogWrite(PWMpin, i);

delay(10);

}

}

void loop()

{

int x = 1;

for (int i = 0; i > -1; i = i + x){

analogWrite(PWMpin, i);

if (i == 255) x = -1; // switch direction at peak

delay(10);

}

}

PARA WHILE

while(expression){

// statement(s)

}

var = 0;

while(var < 200){  
  *// do something repetitive 200 times*  
  var++;  
}

do

{

// statement block

} while (test condition);

#### **Example**

do

{

delay(50); // wait for sensors to stabilize

x = readSensors(); // check the sensors

} while (x < 100);

PARA BREAK

#### **Example**

for (x = 0; x < 255; x ++)  
{  
    analogWrite(PWMpin, x);  
    sens = analogRead(sensorPin);    
    if (sens > threshold){      *// bail out on sensor detect*  
       x = 0;  
       break;  
    }    
    delay(50);  
}

Se continúa por el control de la expresión condicional del bucle, y de continuar con cualquier iteraciones posteriores.

#### **Ejemplo**

de ( x = 0 ; x < 255 ; x ++ )   
{   
    si ( x > 40 && x < 120 ) {      *// crea salto en los valores*  
        continuar ;   
    }   
  
    analogWrite ( PWMpin , x ) ;   
    retardo ( 50 ) ;   
}

para return

int checkSensor(){

if (analogRead(0) > 400) {

return 1;

else{

return 0;

}

}

etiqueta Goto; // Envía el flujo del programa a la etiqueta

#### **Example**

for(byte r = 0; r < 255; r++){

for(byte g = 255; g > -1; g--){

for(byte b = 0; b < 255; b++){

if (analogRead(0) > 250){ goto bailout;}

// more statements ...

}

}

}

bailout:

P16. Escriba e implemente el código que se muestra a la derecha. ¿Para qué sirve el modificador [const](https://www.arduino.cc/en/Reference/Const)?

El **CONST** palabra clave significa constante. Es una variable calificador que modifica el comportamiento de la variable, por lo que una "variable de sólo lectura”. Esto significa que la variable se puede utilizar como cualquier otra variable de su tipo, pero su valor no se puede cambiar. Usted recibirá un error del compilador si intenta asignar un valor a un **const** variable.

const int leD1 = 10; // the number of the pushbutton pin

const int leD1 = 11; // the number of the LED pin

void setup() {

// initialize the LED pin as an output:

pinMode(leD1, OUTPUT);

// initialize the pushbutton pin as an input:

pinMode(leD2, OUTPUT);

}

void loop() {

digitalWrite(leD1,HIGH)

digitalWrite(leD2,LOW)

delay(1000)

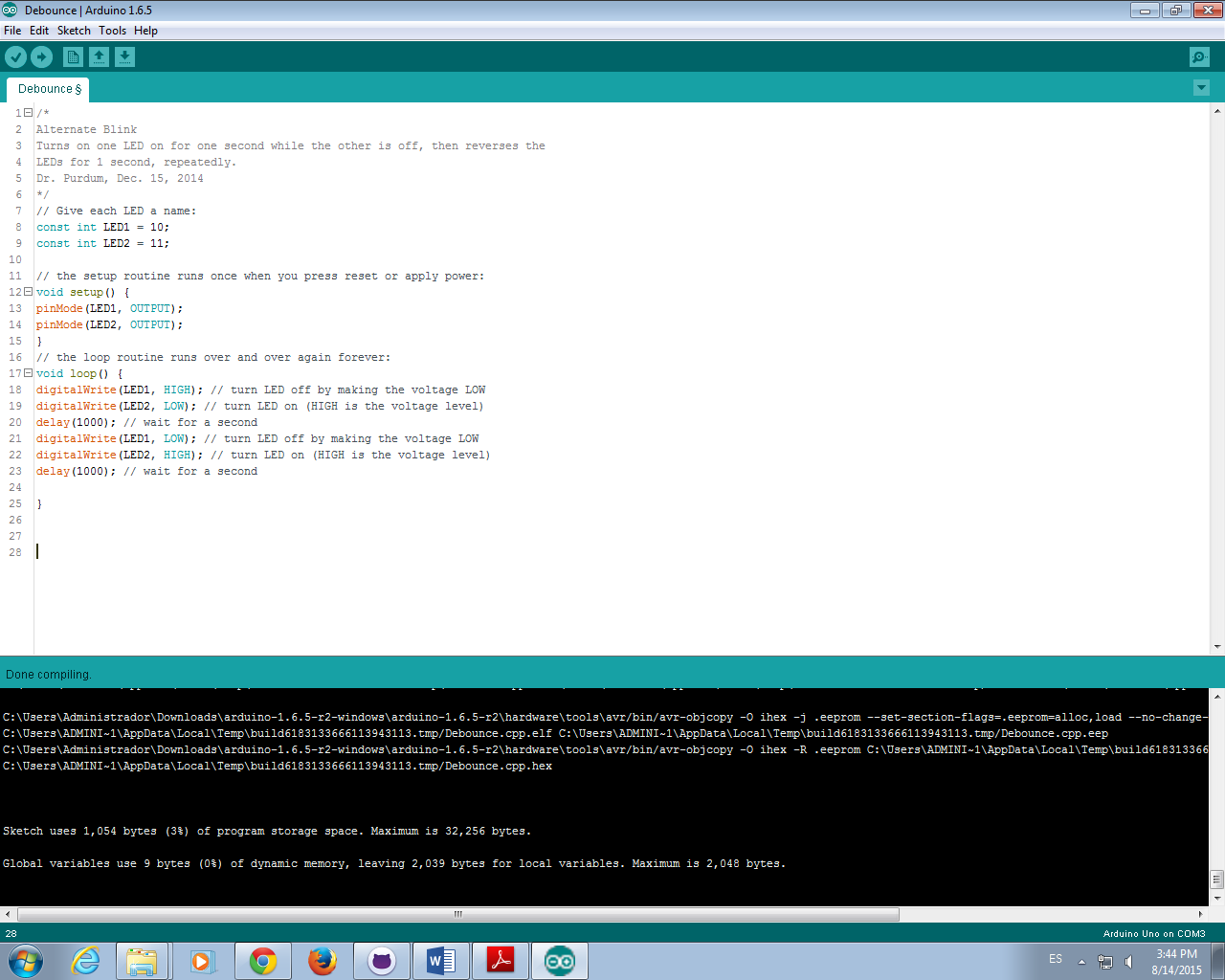
digitalWrite(leD1,LOW)

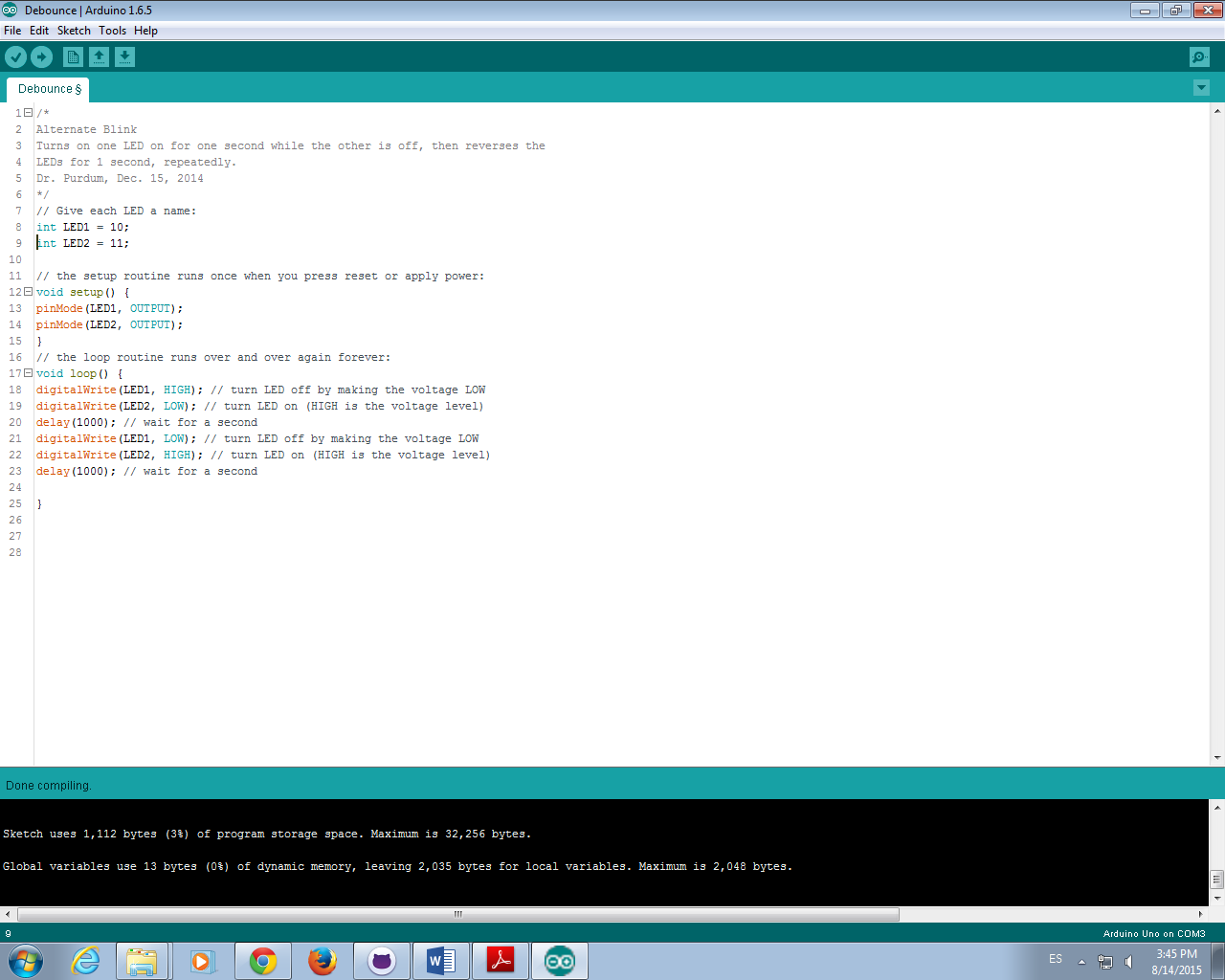
digitalWrite(leD2,HIGH)

delay(1000)

}

P17. Compile de nuevo el programa. Compare el mensaje final del proceso de compilación con el mensaje que aparece en la figura de este slide. ¿Qué se puede concluir?





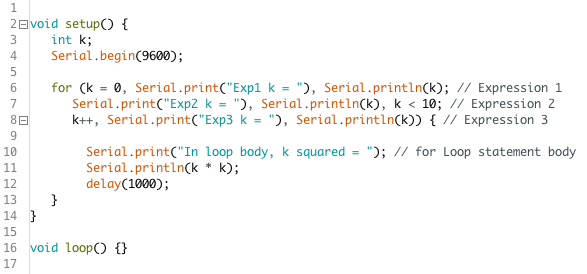
No existe ningún cambio, aparecen los mismos valores

Pg, 78

Implemente el código que se muestra en la figura.

P18. ¿Cómo funciona? explique cada línea de código

P19. ¿Cómo funciona el programa que muestra la figura de la derecha?



El bucle comienza con la definición de variable k, seguido (quizás) por algunas declaraciones adicionales.

A continuación, el bucle se introduce para, expresión1, o k = 0, se procesa. Para esa expresión es normalmente seguida por un punto y coma, **expresión1** es una declaración completa.

La expresión1 puede tener una lista delimitada por comas de sub-expresiones. Por ejemplo, for (k = 0, j = 1; k < 1000; k++) {

En este ejemplo, k variable se define y se inicializa como parte de expresión1.

Siguiendo con la explicación Luego nos muestra el resultado con ‘exp2 k=’. Una vez se procesa la expresión1, el control pasa a expresión2, o k <10

Lo que sucede después depende del resultado de expression2. la expresión2 se evalúa como verdadera lógica, luego se pasa al control de los estados en el cuerpo del bucle

Si expresión2 es verdadero se pasa a las declaraciones en el cuerpo del ciclo luego se procesan y se pasa a l control a expression3 para la evaluación

Si expresión2 es falsa lógica, la defines y el control de bucle pasa a la primera sentencia después de la llave de cierre del bloque de instrucciones de bucle.

Por lo general, expression3 se utiliza para cambiar el estado de la variable que controla las iteraciones de bucle o k variable en nuestro ejemplo.

Después se procesa expression3, el control pasa de nuevo a expresión2 El camino tomado ahora de nuevo depende del resultado de la evaluación de expresión2. Si la expresión se evalúa como verdadera lógica, los estados del cuerpo del bucle se ejecutan de nuevo. Si la declaración se evalúa como falsa lógica, la de fines y el control de bucle se envía a la primera declaración después de la llave de cierre del bloque de declaración.

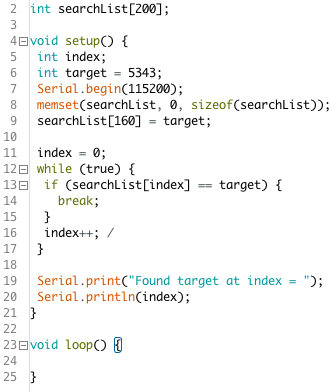
Se puede ver que expresión1 solo es visitado una vez de resto no se vuelve a él.

La segunda línea muestra que expresión2 es visitado siguiente. Debido a que k es menor que 10, que la expresión evalúa a la lógica verdadera, y el cuerpo de bucle se ejecuta. El cuerpo del bucle, simplemente utiliza el objeto de serie al visualizar el cuadrado del valor actual de k.

Ahora observamos cómo ramifican de inmediato a expression3 para cambiar el estado del contador de bucle (es decir, k ++). Una vez que el contador de bucle se cambia (es decir, los cambios de estado de bucle), el código evalúa inmediatamente

expression2 (es decir, k <10) de nuevo para ver si se necesita otra pasada por el bucle. Debido a que k es menor que 10,

P20. Explique el programa que muestra la figura.



Lo primero que hacemos es definir una matriz de 200 int s nombre searchlist []. Debido a que hemos colocado la definición de Lista []

Después de searchlist []., entramos en la ejecución entra en la función setup (), donde se define un par de int, incluimos un nombre en este caso target para n valores de 5343.

También se crean un serial.begin para que podamos usar el monitor de serie y mostrar nuestra salida del programa. (115200)

A continuación, llamamos a la función memset () después ingresamos en ella tres piezas de información (es decir, los argumentos de función):

memset(searchList, 0, sizeof(searchList));

• El nombre de la matriz que queremos inicializar, searchList,

• El valor que queremos inicializar la memoria (es decir, 0)

• El número de bytes que queremos inicializar, sizeof(searchList)

Después de la llamada a memset (), el código establece el elemento 160 de la [] array Lista al valor objetivo de 5343.

Esto significa que searchlist[160] es igual a 5343, mientras que todos los demás valores de 199 se han puesto a 0 por la llamada a memset ().

el siguiente index to 0, que es realmente expresión1 de un bucle de buen comportamiento. Tenga en cuenta que,

el bucle while expresión1 se ha fijado antes de entrar en el código de bucle. Una sentencia if esta comprueba si el elemento de la matriz del código está examinando actualmente (es decir, Lista [índice]) es igual al objetivo.

Si estamos ante uno de los elementos cuyo valor es 0, el caso de prueba falla (lógica falsa) y el código de la si se salta bloque de instrucciones. Esto significa que el índice es post incrementado (index++, que es expression3) y el código se remonta hasta la parte superior del bucle while para comprobar expresión2.

Sin embargo, debido a que escribimos expression2 del bucle while como (verdadero), el programa entra en el bloque de la declaración del bucle while y una vez más se evalúa la expresión if. Este proceso se repite hasta el punto donde el índice es igual a 160.

En nuestro programa, break (salto) envía el control del programa de la primera línea después del bucle while, nos muestra el mensaje y el valor de Índice de donde se encontró el partido. Obviamente, el valor de índice es 160.

P20a. ¿Para qué sirve [memset](http://www.cplusplus.com/reference/cstring/memset/)?

La función memset () es a menudo una pieza de ensamblaje código de idioma ajustado a mano en el compilador para que las personas diseñen e inicialicen un bloque de memoria a un valor específico. (La función memset () es parte de la colección de bibliotecas de funciones que vienen con el Arduino IDE). Los tres argumentos de la función que van allí son Listas [] array, un valor específico (es decir, 0). Sin embargo, no hemos visto el operador sizeof () antes.

**Rellene bloque de memoria**

Establece los primeros *num* bytes del bloque de memoria apuntado por *ptr* a la especificada *valor* (interpretado como ununsigned char ).

P20b. ¿Para qué sirve sizeof?

sizeof () es un operador que devuelve el Número de bytes asignados a un elemento de datos.

P21c. ¿Qué valor devuelve sizeof(searchList)?

El número de bytes que queremos inicializar

Debido a que hemos puesto el nombre de la matriz BúsquedaLista dentro los paréntesis del operador sizeof (), la evaluación de esa expresión devuelve el número 400.

P20d. ¿Cuál es la función de la palabra reservada break en la línea 14?

Break (salto): envía control del programa a la primera línea después del bucle while, que hace una llamada al objeto de serie para mostrar un mensaje y el valor deÍndice de donde se encontró el partido. Obviamente, el valor de índice es 160.