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.

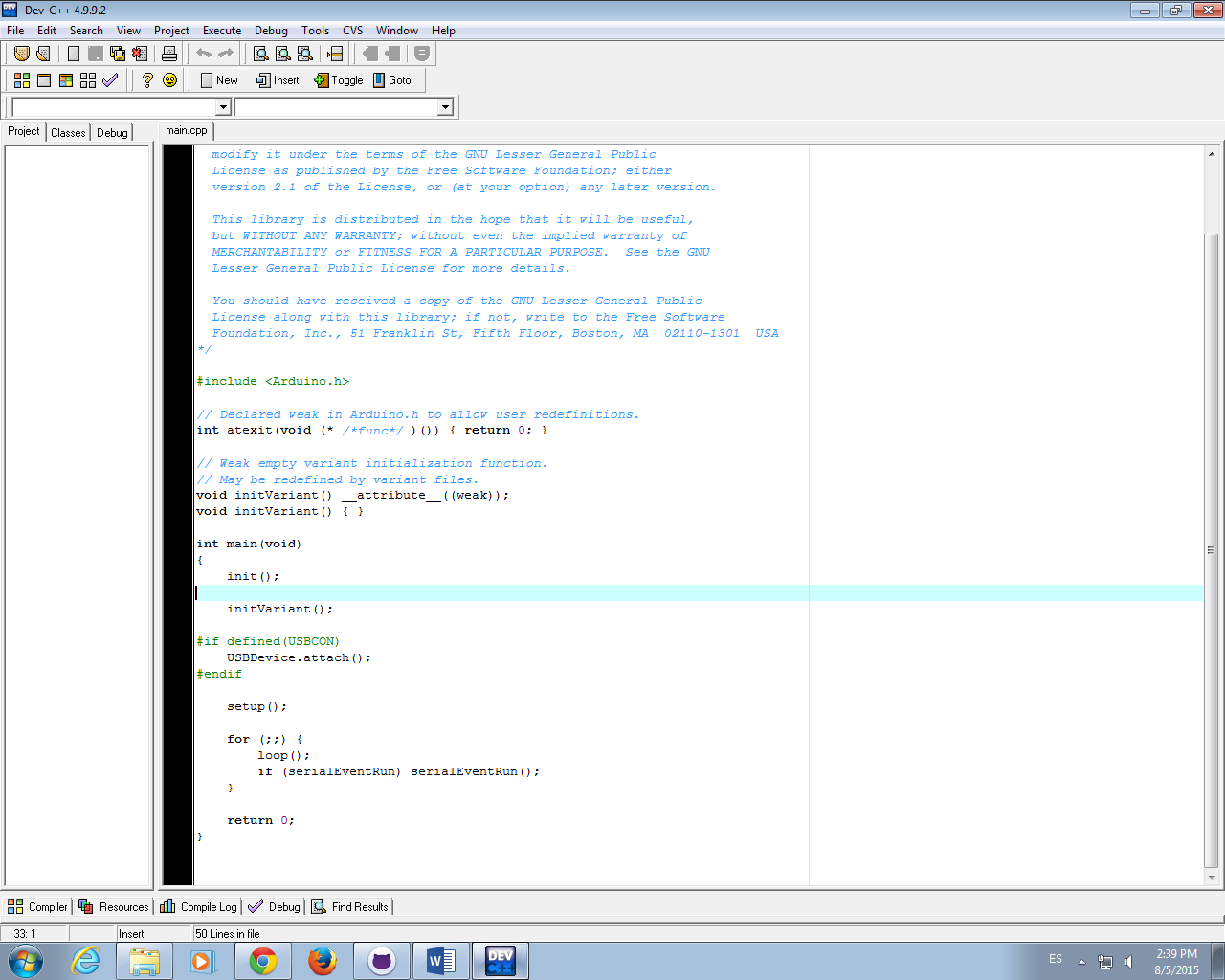
* P3: ¿Cuál es la [estructura](https://www.arduino.cc/en/Reference/HomePage) de un programa para un Arduino?
* [setup](https://www.arduino.cc/en/Reference/Setup)()
* [loop](https://www.arduino.cc/en/Reference/Loop)()
* P4: ¿Cuál es la función de cada una de las partes de la estructura de un programa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

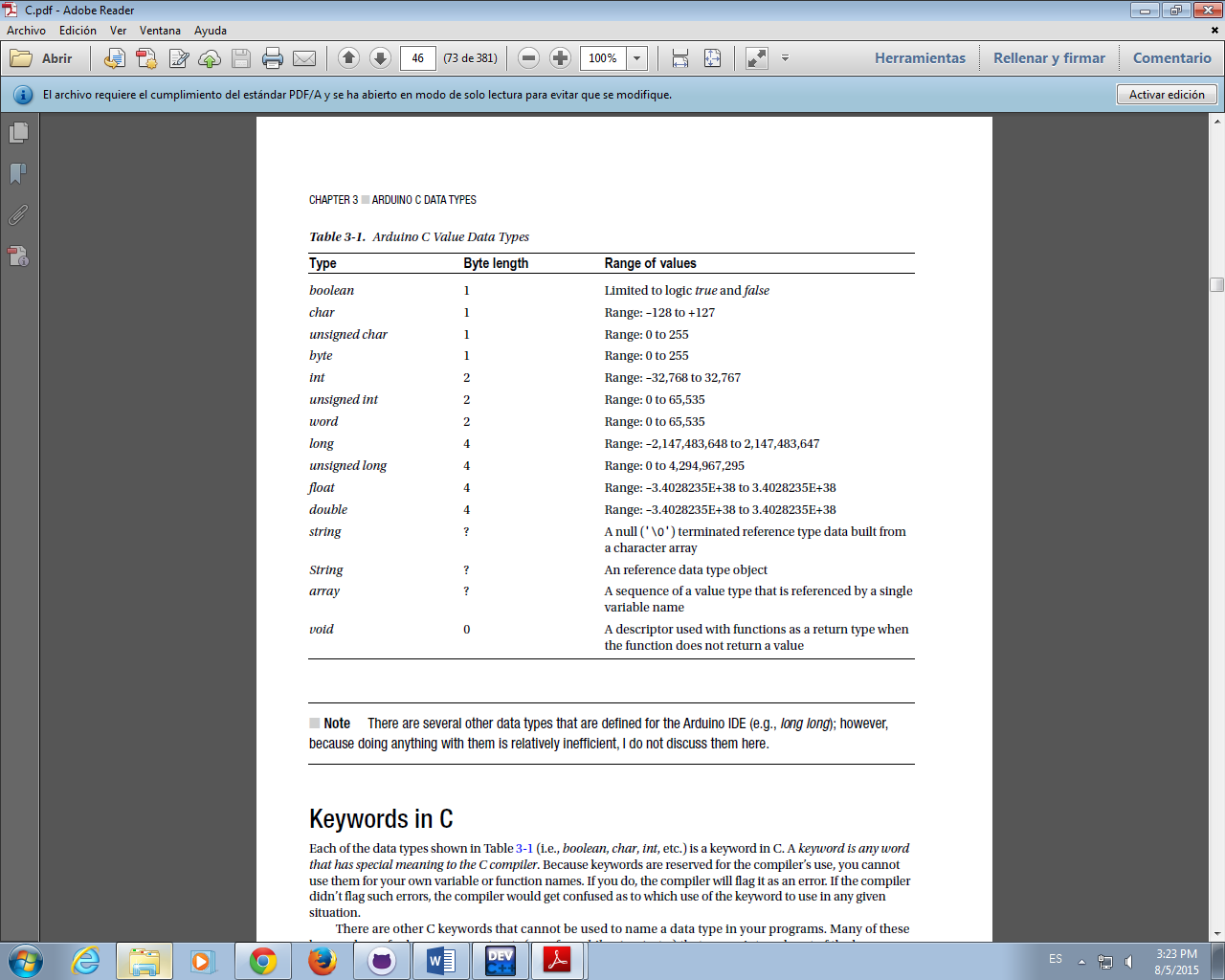
P5b. ¿Cuántas veces se ejecuta la función loop?

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.



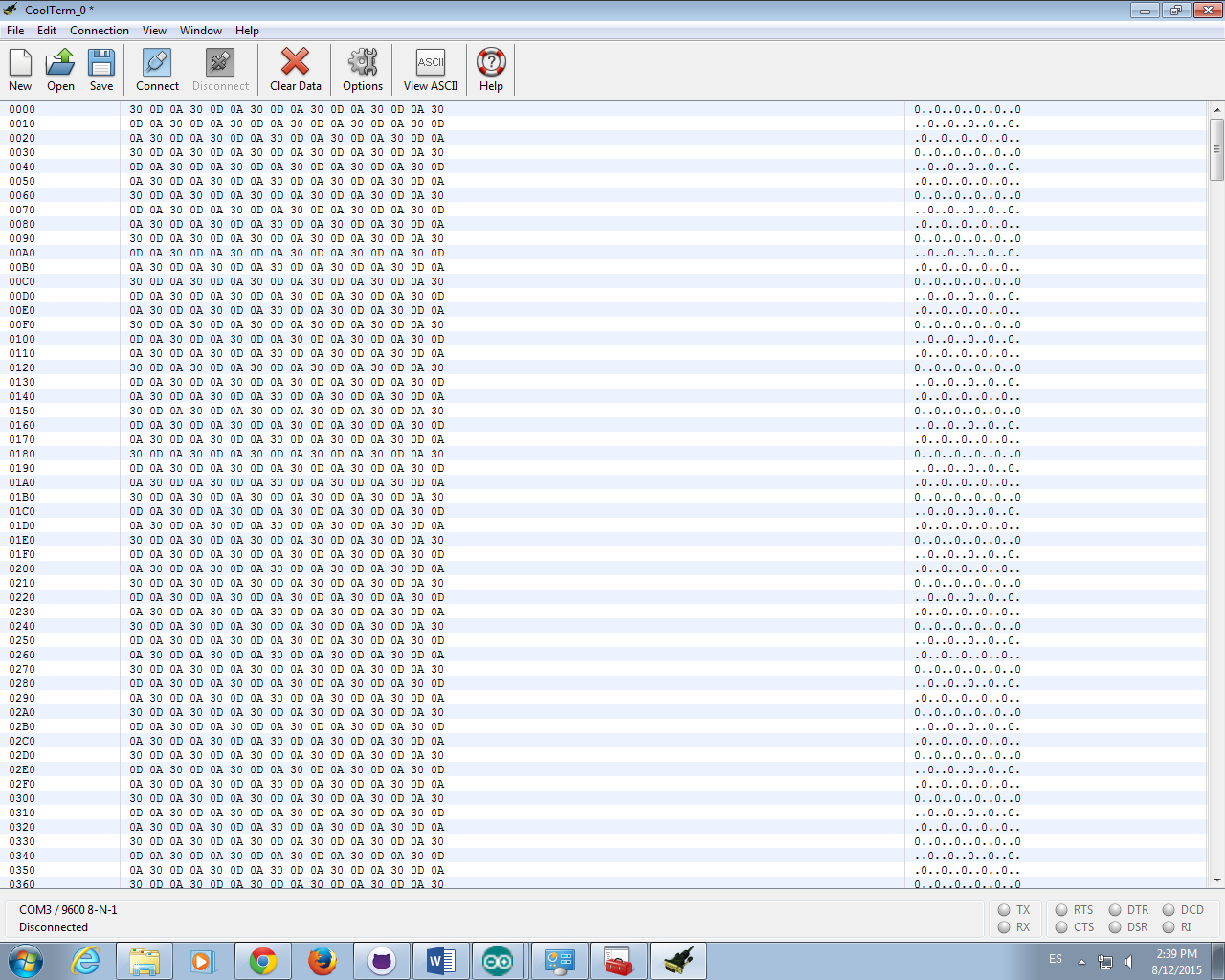
P7. ¿Para qué sirve terminar un [string](https://www.arduino.cc/en/Reference/String) en null (código [ASCII 0](http://www.asciitable.com/))?

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:

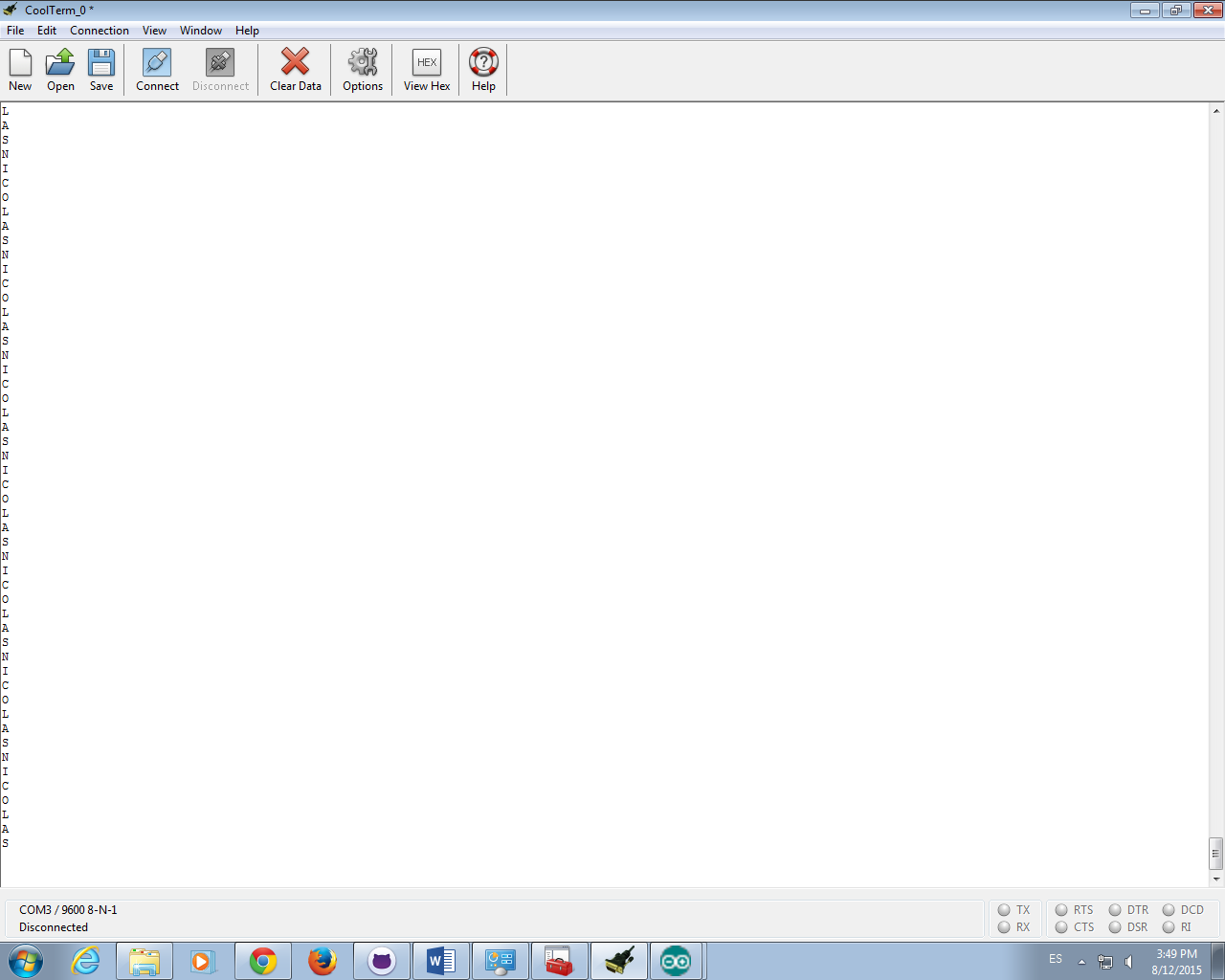
int i;

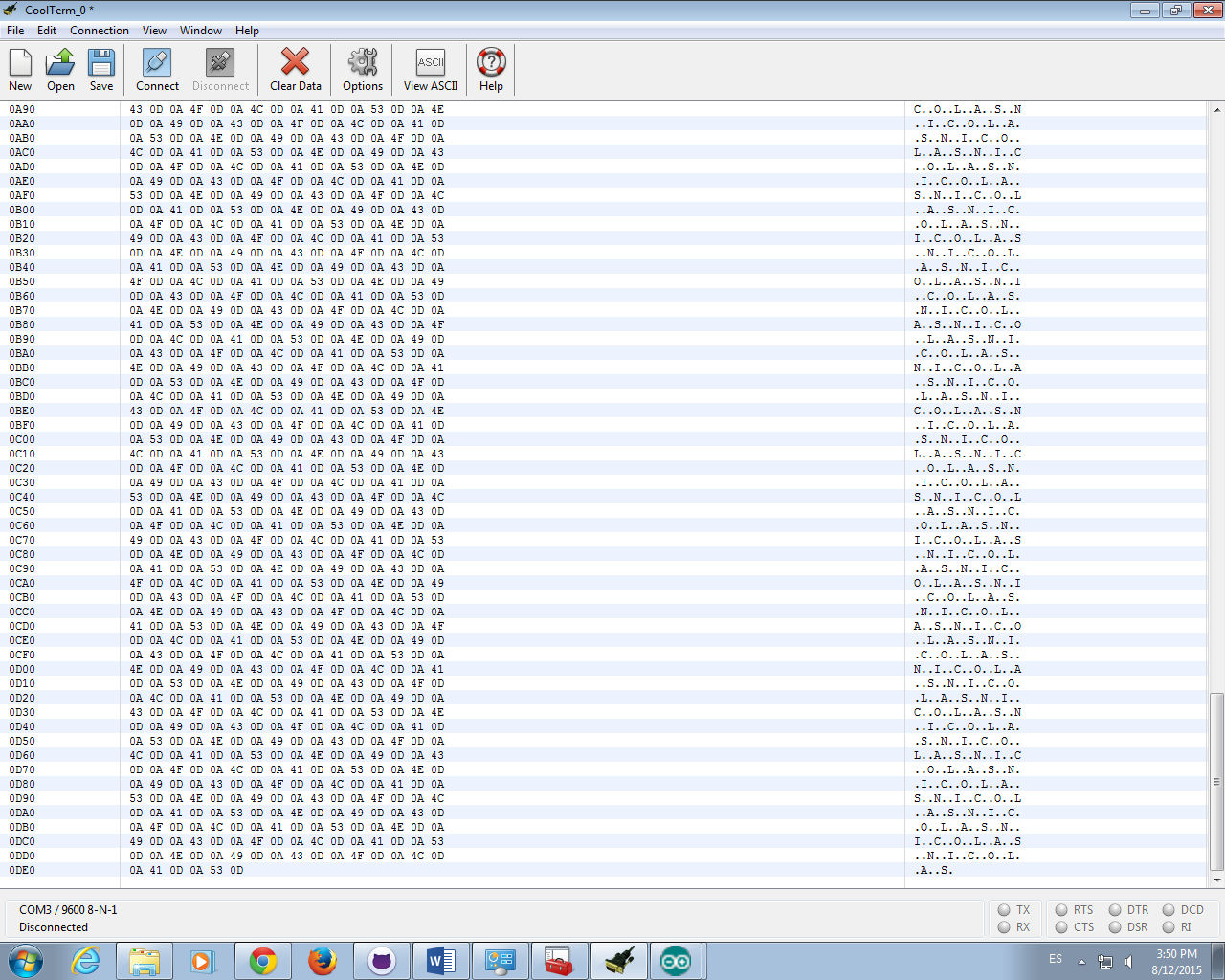
for (i = 0; i < 7; i = i + 1) {

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)

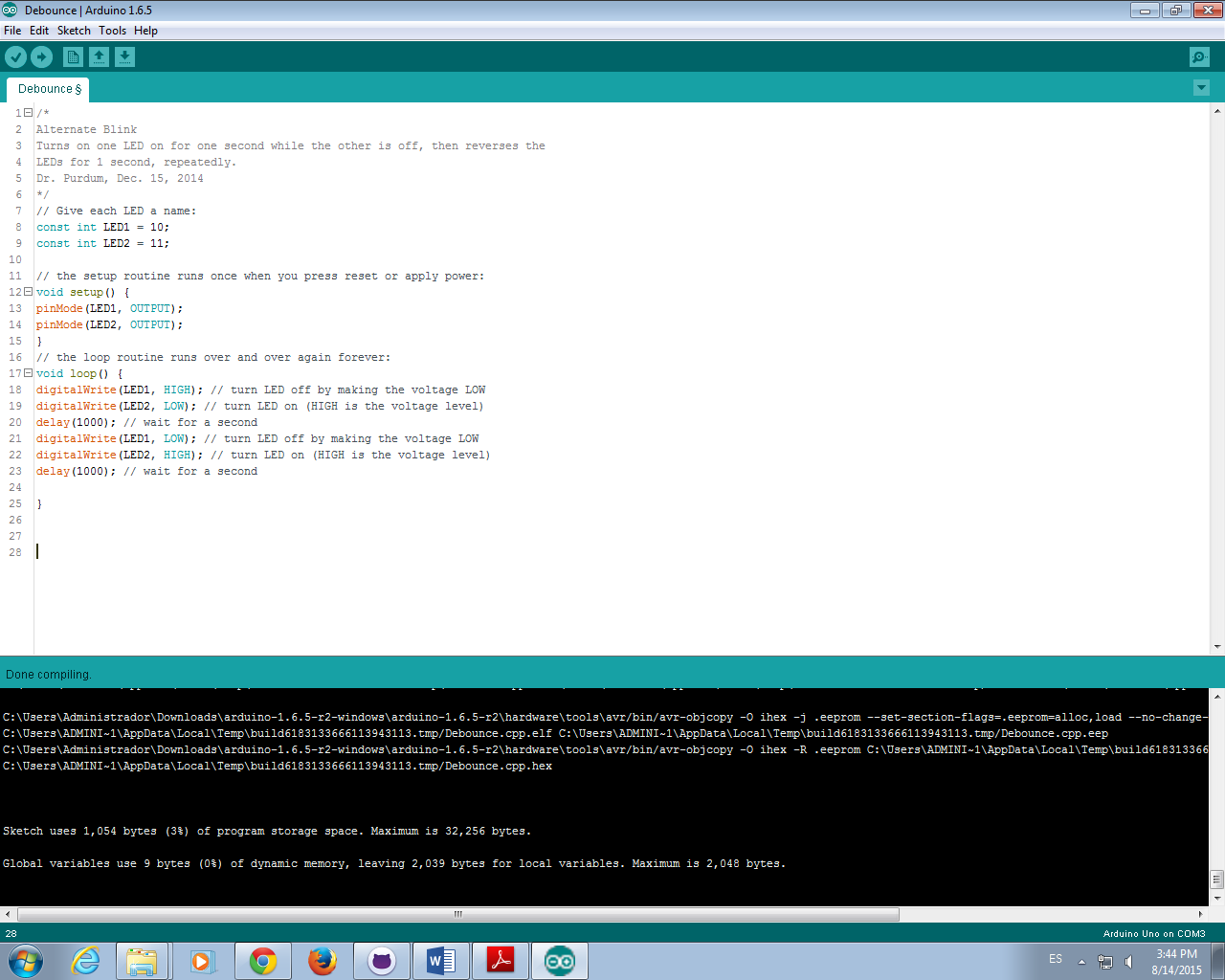
digitalWrite(leD1,LOW)

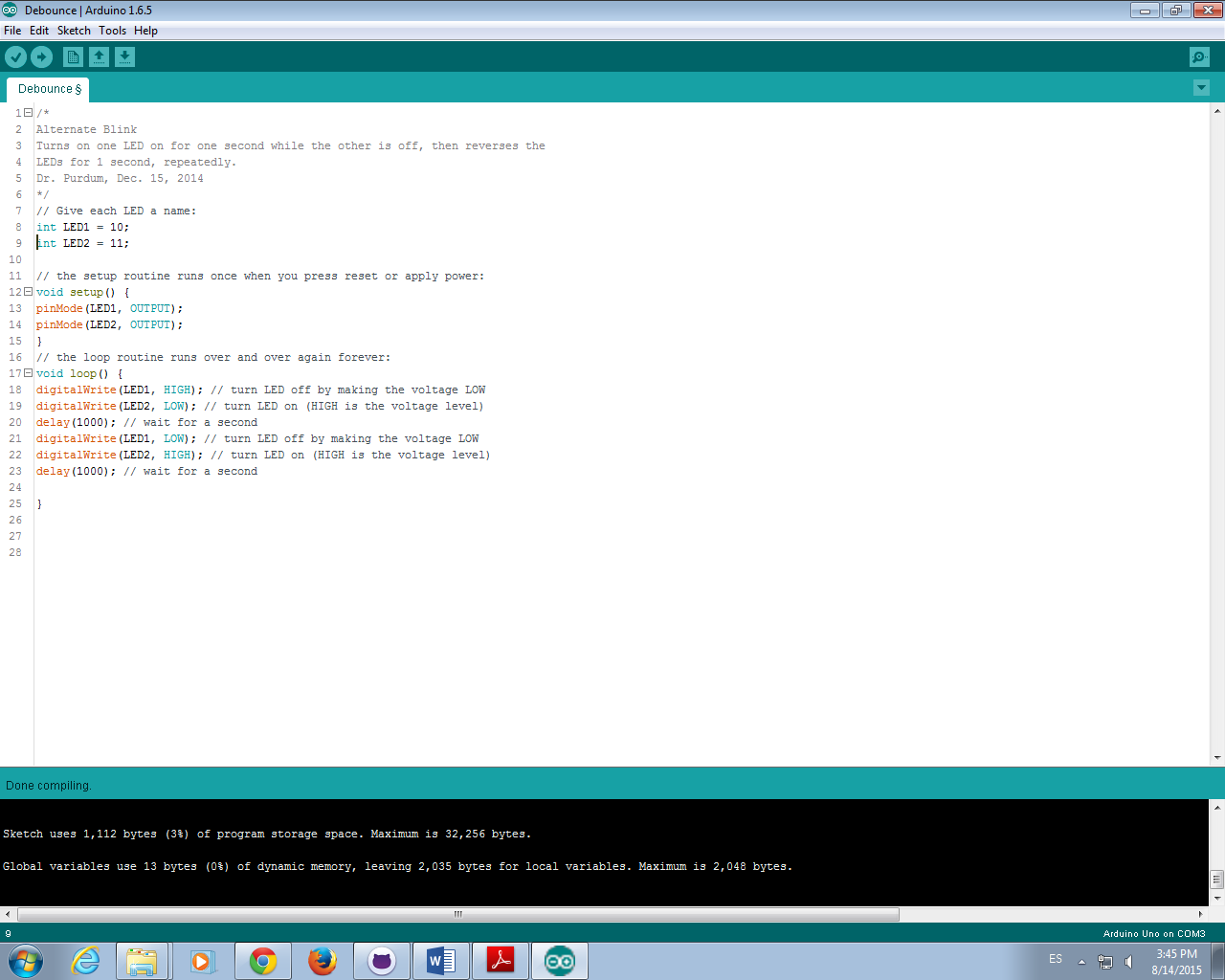
digitalWrite(leD2,HIGH)

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?

P20. Explique el programa que muestra la figura.

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

P20b. ¿Para qué sirve sizeof?

P21c. ¿Qué valor devuelve sizeof(searchList)?

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