Nombre del proyecto: Parkimaniacos

Código Arduino

Equipo de trabajo:

José Ángel García Arce	NC: 17130786
Luis Emmanuel Méndez Barrios	NC: 17130804
Enrique Antonio Belmarez Meraz	NC: 17130765
José Baltazar Martínez De La Rosa	NC: 17130049

CODIGO:

```
//Incluimos las librerias necesarias para la ejecucion del programa
#include <SPI.h>
#include <UIPEthernet.h>
//Declaramos todas las variables globales que se utilizaran
  String idparq = "1";
  byte mac_addr[] = { 0x74,0x69,0x69,0x2D,0x30,0x31 };
  long t; //timepo que demora en llegar el eco
  long d; //distancia en centimetros
  int x = 0; // variable para auxiliar en el tiempo de Scaneo de QR
  boolean bandera = false; //Bandera para no estar haciendo peticion cada ciclo del loop , sino
cada cambio de estado
  boolean QR = false;
  EthernetClient cliente; //Declaracion del modulo de Ethernet en modo cliente
  const int Trigger = 8,Echo = 9, ledVerde= 2, ledRojo = 4, ledAmarillo = 3; //Declaracion de los
Pines, Pin digital 8 para el disparo del ultrasonico, 9 para el echo , 2 para el led disponible y
3 para el led de ocupado

//void setup
void setup() {
  pinMode(Trigger, OUTPUT); //pin como salida
  pinMode(Echo, INPUT);  //pin como entrada
  pinMode(ledVerde, OUTPUT); //pin como salida
  pinMode(ledRojo, OUTPUT); //pin como salida
  pinMode(ledAmarillo, OUTPUT); //pin como salida
  digitalWrite(Trigger, LOW);//Inicializamos el pin con 0

  Serial.begin(9600); //Inicializamos el Serial con los Baudios 9600
  //condicion para hacer conexion al modulo Ethernet enc28j60
  if(!Ethernet.begin(mac_addr))
    Serial.println("la conexion a controlador fallo");
   else{Serial.println("la conexion a controlador fue un exito");
    }
}

//void loop
void loop() {

  if(cliente.connect( "maniacorp.ddns.net" , 3000 )){ //condicion donde se hace la conexion al
servidor
  digitalWrite(Trigger, HIGH); // ponemos el pin de disparador en 1 para que mande la onda sonica
  delayMicroseconds(10);//damos un delay para que mande la onda sonica
  digitalWrite(Trigger, LOW);   // ponemos el pin de disparador en 0 para que deje de mandar la
onda sonica
  t = pulseIn(Echo, HIGH);     // ponemos el pin de eco en 1 para que reciba la onda sonica
  d = t/59;  //formula para calcular la distancia
  Serial.println(d); //en esta linea es para mostrar al serial la distancia
  if(d>100){  //Condicion cuando la distancia sea mayor a 1 metro
    if(bandera==false){ //condicion para que no se envien los cada ciclo
      //enviando datos
        mandardis(idparq ,"0"); //llamamos al metodo para mandar la disponibilidad del
parquimetro con un 0 como disponible
          digitalWrite(ledAmarillo, LOW); //Apagamos el led amarillo
          digitalWrite(ledRojo, LOW); //Apagamos el led Rojo
          digitalWrite(ledVerde, HIGH); //Prendemos el led Verde para indicar que esta disponible
el Parquimetro
        bandera = true; //ponemos el true la bandera para solo mandar 1 vez la peticion
    }
  }
  else{
     if(bandera== true){ //condicion para que no se envien los cada ciclo
       //enviando datos
        mandardis( idparq ,"1"); //llamamos al metodo para mandar la disponibilidad del
parquimetro con un 1 como Ocupado
        digitalWrite(ledAmarillo, HIGH); //Prendemos el led Rojo Indicando que esta ocupado el
parquimetro
        digitalWrite(ledRojo, LOW);   //Apagamos el led Rojo
        digitalWrite(ledVerde, LOW);  //Apagamos el led verde
        while(true){ //mandar peticiones hasta que el qr sea leido
          //condicion para checar si el QR del parquimetro fue scaneado o no
```

```
          if(checarQR( idparq )){
            QR = true;  // Hacemos que QR sea verdadero
            break;  //teminamos el ciclo while
           }
          else{ // Parte contraria de la condición del chechado del QR
            QR = false; // Hacemos que QR sea falso
            x++;
            if(x > 5){ // condición para crear un tiempo entre el estacionado y la lectura del QR
              x=0;    //hacemos que la variable del axuliar de tiempo sea 0
              break;   //teminamos el ciclo while
              }
          }
          delay(3000); //esperamos un lapso de 3 segundos para volver a hacer la peticion
          }
        if(QR){
          digitalWrite(ledAmarillo, LOW); //Apagamos el led amarillo
          digitalWrite(ledRojo, HIGH);   //Prendemos el led Rojo Indicando que esta ocupado el
parquimetro y a su vez tambien se scaneo el QR
          digitalWrite(ledVerde, LOW);  //Apagamos el led verde
        }
        bandera = false;  //ponemos el false la bandera para solo mandar 1 vez la peticion
(Intercalando los 2 estados)
    }
  }
  }
  else {
    Serial.println("conexion fallida al servidor"); //Mandamos al serial cuando falla la conexion
al servidor
    cliente.stop(); //en cao de tener falla al conectar con el servidor detenemos el cliente

  }
  delay(500); //damos un delay minimo de apoyo
  cliente.stop(); //Cerramos la conexion del cliente para poder volver a iniciar la conexion al
principio del loop
}

//---------------METODOS----------------------------
//Metodo para mandar la peticion de disponibilidad al servidor
void mandardis(String id , String disp){
        cliente.print("GET /actualizar?id="+id+"&estado="+disp); //Se usa una funcion del cliente
llamada print para buscar en el servidor la ruta
        finaldelinea(); //metodo para reusar codigo extra que se manda en cada peticion
        Serial.println("Se modifico con exito la disponibilidad:\n con id= "+id+"\n y un dato de=
"+disp); //mandamos al serial el cambio que se hizo
  }

//Metodo para mandar la peticion de Checar si se Escaneo el QR
boolean checarQR(String id ){
      cliente.connect( "maniacorp.ddns.net" , 3000); //hacemos la conexion por si llega a fallar
la conexion a mitad de ciclo
      cliente.print("GET /checar?id="+id); //Se usa una funcion del cliente llamada print para
buscar en el servidor la ruta
      finaldelinea();  //metodo para reusar codigo extra que se manda en cada peticion
      char c ;  //declaramos un caracter
      String codigo = "",aux=""; //declaramos 2 String auxiliares
      while(true){ //ciclo while para poder recorrer todo los caracteres recibidos del servidor
        if(cliente.available()) { //condicion si el cliente esta disponible para leer
          c = cliente.read(); // guardamos el en caracter lo que en el servidor le manda
          codigo = codigo + c; //agregamos caracter por caracter a la string codigo para guardar
todo en esa variables
          //Serial.print(c); //Mandamos al serial el caracter para ver si esta recibiendo y que
esta recibiendo
        }
       if (!cliente.connected()) { //se hace la condicion para cuando acabe de mandar todos los
caracteres
        Serial.println();
        Serial.println("disconnecting."); //mandamos al Serial un aviso de desconexion
        //cliente.stop();
        break; //ponemos un break para cuando se termine todo salga del ciclo while
        Serial.print(codigo); //// mostramos el código que hemos obtenido de la web.
       for(;;)
```

```
          ;
        }
      }
      //Metemos en la variable auxiliar los ultimos 4 caracteres de la variables codigo para
quitar todo el encabezado que manda el servidor
      aux = codigo.substring(codigo.length()-4,codigo.length());
      Serial.println("\n o el codigo es: "+ codigo + "\n o el aux es: "+ aux); //mandamos al
Serial la variable codigo y la varibale aux y notar la diferencia
        if(aux == "true"){ //condicion para comparar si la variable aux es un true
          Serial.println(" entro al true"); //mandamos al serial un aviso  de que es true
          return true;   //Regresamos un true
        }
      if(aux == "false"){ //condicion para comparar si la variable aux es un false
        return false;  //Regresamos un false
        }
      return false;  //en caso de no entre a ningun if mandamos por defaul un false
  }
//metodo para mandar a las peticion que reconozca el servidor de que se esta pidiendo desde un
arduino
void finaldelinea(){
     cliente.println(" HTTP/1.0");
     cliente.println("User-Agent: Arduino 1.0");
     cliente.println();
  }
```

Anexo:

Para programar un Arduino, el lenguaje estándar es C++, aunque es posible programarlo en otros lenguajes. No es un C++ puro sino que es una adaptación que proveniente de avr-libc que provee de una librería de C de alta calidad para usar con GCC en los microcontroladores AVR de Atmel y muchas funciones específicas para los MCU AVR de Atmel.

avr-binutils, avr-gcc y avr-libc son las herramientas necesarias para programar los microcontroladores AVR de Atmel.

El *software* de Arduino consiste de dos elementos: un entorno de desarrollo (IDE) (basado en el entorno de *processing* y en la estructura del lenguaje de programación Wiring), y en el cargador de arranque (*bootloader,* por su traducción al inglés) que es ejecutado de forma automática dentro del microcontrolador en cuanto este se enciende.