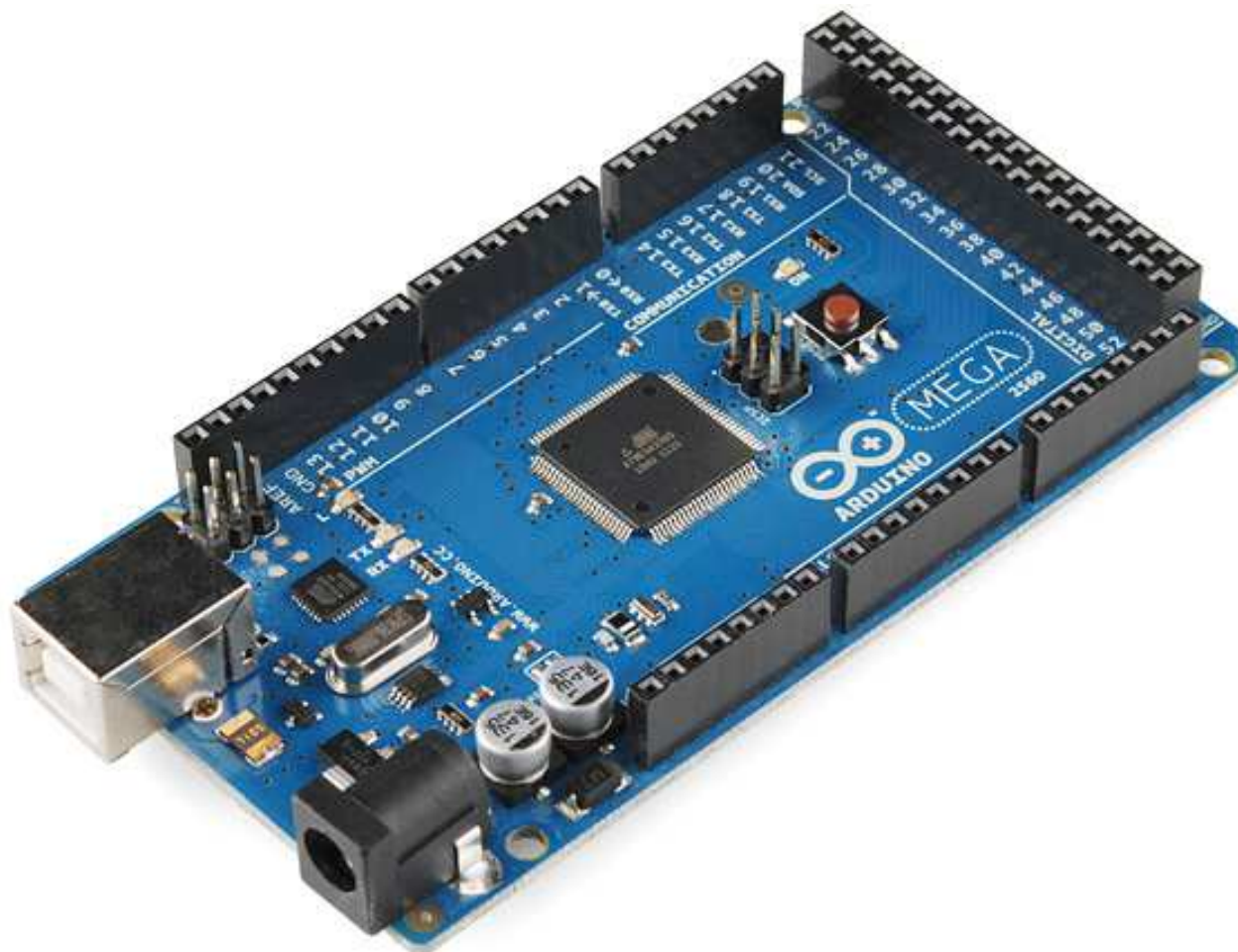


En estas páginas, daremos juntos los primeros pasos en la programación y construcción de circuitos, robots, si...!! Robots diseñados y contruidos con nuestras manos..!!!.

Todos los proyectos que acá encontrarás, fueron Programados con el IDE de Arduino, Versión 1.8.5 y probados en una placa Arduino MEGA 2560 (Aunque se puede usar cualquiera de los modelos Arduino encontrados en el mercado)



ESPECIFICACIONES

- Procesador: ATmega2560
- Voltaje de operación: 7V-12V
- Entradas/Salidas Digitales: 54 (14 programables como salidas PWM)
- Entradas Analógicas: 16
- Capacidad de corriente por pin: 40mA
- Memoria Flash: 256kB (4kB son empleados por el bootloader arduino)
- Memoria RAM: 8kB
- Memoria EEPROM: 4kB
- Velocidad de Clock: 16Mhz

La intención de la presente, es que el estudiante analice, pruebe y entienda todos y cada uno de los proyectos acá desarrollados.

Avanzando según el interés de cada uno.

Para esto dispone además de clases guiadas.

Comencemos: Llamaremos Robot a una entidad virtual o mecánica, artificial, automática y programable, capaz de realizar determinadas operaciones de

manera autónoma. Puede estar dotada de sensores, que le permiten adaptarse a situaciones.

La **robótica moderna** difiere de la antigua, ante el lógico avance científico. Sin embargo, los primeros intentos de crear robots se remontan al siglo IV a.C., cuando el matemático griego **Arquitas de Tarento** logró construir un ave mecánica que funcionaba con vapor.

Arduino, es una compañía open source y open hardware, así como un proyecto y comunidad internacional que diseña y manufactura placas de desarrollo de hardware para construir dispositivos digitales y dispositivos interactivos que puedan censar y controlar objetos del mundo real. Arduino se enfoca en acercar y facilitar el uso de la electrónica y programación de sistemas embebidos en proyectos multidisciplinarios.

LISTADO DE PROYECTOS

GUIA 01 – (Usa Lista de Materiales 01)

1	Prendido y apagado de un LED. Programa y circuito, que prende y apaga un LED a intervalos regulares.
2	Prendido y Apagado de un LED – Esta Vez Usando variables. Modificación del Programa Anterior. Programa y circuito, que prende y apaga un LED a intervalos regulares.
3	Prender 2 LED – Alternando el prendido y apagado. Pender dos LED alternándolos cada medio segundo - Cuando uno se prende el otro se apaga.
4	Prender 3 LED en Secuencia y Luego Apagarlos en la misma Secuencia.
5	Informar en el Puerto Serie Cuando se Prende o Apaga un LED. Hacer un programa y circuito, que prenda y apague un LED y se informe en la Pantalla de la PC (por medio del Puerto Serie) cada vez que el Led se prenda o apague.
6	Prender 3 LED en Secuencia y Luego Apagarlos en la misma Secuencia - Informando las acciones por el Puerto Serie. Informa a través del puerto serie, en que puertos se han configurados los LEDs y cada vez que se prenda o apague un Led.
	Conociendo el "if()" - Parte I - Informar a través del Puerto Serie, si una variable contiene un valor mayor o igual a 10.
	Conociendo el "if()" - Parte II - Informar a través del Puerto Serie, El mayor valor contenido en las Variables "Valor_1" y " Valor_2"
	Contando Milisegundos - Mostrar a través del Puerto Serie los milisegundos transcurridos desde el momento que se prende nuestra placa Arduino
	Contando Segundos - Hacer un programa que nos permita contar y mostrar a través del Puerto Serie, los segundos transcurridos desde que se prendió nuestra placa Arduino.
	Contando Minutos y Segundos - Hacer un programa que nos permita contar y mostrar a través del Puerto Serie, los Minutos y Segundos transcurridos desde que se prendió nuestra placa Arduino.
	Controlar tiempo - Enviar mensaje "Hola" cada 5 segundos a través del Puerto Serie.
7	Prender y Apagar un LED sin usar la función “delay()”. Este programa es una variante de otro que ya construimos anteriormente, pero es muy importante, ya que controlamos el

	tiempo sin la función “ delay() ”. Esta forma de programar la pondremos en práctica siempre que sea posible, ya que es el primer paso a la Programación Multitarea.
8	Prender 2 LED – Alternando prendido y apagado. Sin usar Función “delay()”. Prender dos LED alternándolos cada medio segundo - Cuando uno se prende el otro se apaga. Este programa es una variante de otro que ya construimos anteriormente, pero es muy importante, ya que controlamos el tiempo sin la función “ delay() ”. Esta forma de programar la pondremos en práctica siempre que sea posible, ya que es el primer paso a la Programación Multitarea.
9	Prendido y apagado de un LED en Forma Gradual (Variando la intensidad). Programación Multitarea. Programa y circuito electito que prende gradualmente un LED, subiendo la intensidad desde su mínimo hasta llegar al máximo, luego lo apague gradualmente, desde el máximo hasta su mínimo.
10	Semáforo Simple. Este programa es muy parecido al de los semáforos reales, que solo controlan las luces.
11	Semáforo Simple. Programación Multitarea - Permitiendo otras tareas simultaneas. Implementar un Semáforo y permitir al programa, hacer otras cosas mientras el semáforo esta trabajando.
12	Semáforo para Autos y Peatones. Programación Multitarea. Implementar un Semáforo, sin usar función “ delay() ”. El programa debe poder hacer otras cosas mientras el semáforo esta trabajando.
13	Interactuando con la PC – (Primera Parte) - Dar la orden desde la PC para prender o apagar un LED (Puerto Serie). Programación Multitarea. Enviar comandos desde la Computadora que permitan Seleccionar, Prender o Apagar un LED entre tres. En caso de recibir un Comando Erróneo, el programa debe informar mediante un mensaje por el puerto Serie.
14	Interactuando con la PC – (Segunda Parte) - Dar la orden desde la PC para prender o apagar un LED (Puerto Serie). Ejemplo estructura switch. Este ejercicio, ya lo habíamos resuelto, pero usando “ if ”. Enviar comandos desde la Computadora que permitan Seleccionar, Prender o Apagar un LED entre tres. En caso de recibir un Comando Erróneo, el programa debe informar mediante un mensaje por el puerto Serie.
15	Como detectar desde Arduino, cuando tenemos DATOS para leer que ingresan por el puerto Serie. (la PC envía, Arduino recibe)
16	Recepción De Números Enteros - la PC envía, Arduino recibe.
-	Apéndice A - Como usar una Placa PROTOBOARD.

GUIA 02 – (Usa Lista de Materiales 01)

1	Reconocimiento del Estado de un Botón Pulsador (Cuando es Presionado). Programación Multitarea.
2	Prender un LED cuando un Botón Pulsador es Accionado (Primera Parte). En este ejemplo, prenderemos un LED cuando detectemos que un interruptor ha sido presionado (accionado). Dicho en otras palabras, cuando el botón es presionado, y mientras este presionado , se prendera el LED.
3	Prender un LED cuando un Botón Pulsador es Accionado (Segunda Parte). Haremos una variación del ejercicio anterior, esta vez, en caso de que el interruptor este dejando pasar corriente (el sistema esta Activo), se prenda un LED avisando que TODO ESTA BIEN Y OPERANDO. De lo contrario, otro LED avisando que El circuito no esta conectado (Inactivo).
4	Encender y apagar un LED con el mismo botón pulsador. Programa que permite detectar cuando un Botón Pulsador es presionado, y mediante esa detección prender y apagar un LED.
5	Encender y Apagar un LED por medio de 3 botones (en forma indistinta). El

	programa permite prender o Apagar un LED por medio de 3 botones (prende desde cualquiera y apaga desde cualquiera de ellos).
6	Aumentar/Reducir intensidad LED con 2 Botones. Este programa permite Subir y/o Bajar la intensidad con que brilla un LED. Botón para Subir y Botón para Bajar.
7	Semáforo (02) inteligente de una sola calle. Semáforo para Autos y Peatones. Funciona Permanentemente, pero mediante botón detiene a los autos y acciona semáforo de Peatones.
8	Simulación de un dado al ser lanzado. Nuestro circuito, simulará la cara de un DADO, donde cada LED será uno de los puntitos que nos muestra la cara del dado.
9	Prender y Apagar Gradualmente un LED (Subiendo su brillo desde cero a Máximo y desde el Máximo a cero). Uso de la Estructura Repetitiva “for”.
10	Prender y Apagar 6 led en secuencia, como Luces de navidad. Uso de la Estructura Repetitiva “for”.
11	Selección de opciones, entre 5 secuencias distintas. Mediante el botón seleccionar una de 5 secuencias rítmicas con los LEDs. En este programa, podrás ver, que pasa cuando usas “ <code>delay()</code> ” en bucles que solo hacen el bucle, es decir, no utilizas la programación multitarea. Cuando este proyecto este funcionando, veras, que en algunas oportunidades, debes mantener apretado unos segundos el botón para que el programa te detecte. Mas adelante, veremos como solucionar este inconveniente.

GUIA 03 – (Usa Lista de Materiales 02)

1	Reconocimiento de un Potenciómetro - Lectura de Valores Arrojadados. Lecturas de los valores entregados por un potenciómetro y conversión a valor porcentual mediante la función “ <code>map()</code> ”. Visualización de Resultados en el monitor del Puerto Serie.
2	Desde un Potenciómetro, aumentar o disminuir la intensidad de un led. Tomar los valores entregados por un potenciómetro y usarlos para aumentar o disminuir la intensidad de un led.
3	Controlar la Frecuencia de parpadeo de un led con un potenciómetro. Con un potenciómetro controlar el tiempo de espera en prender y apagar.
4	Medir Potencia que arroja el Potenciómetro, en forma visual (Vúmetro usando 6 Leds). Medir Potencia liberada por un potenciómetro, con una escala de 6 Leds. Recordar que los valores del potenciómetro oscilan entre 0 y 1023. Incluir Visualización de Resultados en el monitor del Puerto Serie.
5	Leer Dos Potenciómetros Simultáneamente y Mostrar ambos Valores por el Puerto Serie. La idea de este ejercicio es mostrar como leer dos sensores o instrumentos de control simultáneamente y procesarlos separadamente (haciendo dos tareas distintas).
6	Emitir Beep-Beep por el Speaker/Zumbador - (Versión A) - Usando “<code>delay()</code>”. Este programa emite Beep-Beep Intermitente por el Speaker/Zumbador conectada a la placa Arduino. Veremos a continuación, las dos formas de hacerlo, primero usando las Funciones “ <code>tone()</code> ” y “ <code>noTone()</code> ”. Luego con la ya conocida instrucción “ <code>analogWrite()</code> ”.
7	Emitir Beep-Beep por el Speaker/Zumbador- (Versión B) – Programación Multitarea, sin usar <code>delay()</code>. Estos programas, al igual que los dos anteriores, emiten Beep-Beep Intermitente por el Speaker/Zumbador conectada a la placa Arduino, pero sin usar la función “ <code>delay()</code> ”. Primero usando las Funciones “ <code>tone()</code> ” y “ <code>noTone()</code> ”. Luego con la ya conocida instrucción “ <code>analogWrite()</code> ”.
8	Timbre. Activar Timbre cuando Botón Pulsador es Accionado. En este ejemplo, simulamos el Timbre común que tenemos en casa. Dicho en otras palabras, cuando el botón es presionado, y mientras este presionado, suena el timbre.
9	Encender y Apagar un Timbre por medio de 3 botones (en forma indistinta). Con Programación Multitareas. El programa permite prender o Apagar un Timbre por

	medio de 3 botones (prende desde cualquiera y apaga desde cualquiera de ellos).
10	Generando sonido con un potenciómetro. La idea de este ejercicio es generar diversos tonos en el Parlante/Buzzer a partir del estado analógico del potenciómetro.
11	Mini Piano Digital. Con Programación Multitareas. El programa permite componer melodías con ocho notas: 'c', 'd', 'e', 'f', 'g', 'a', 'b' y 'C'. El presente es un simple ejemplo de creación de música.
12	Recorrido de una octava. – Idea para los que entienden de Música. Almacenaremos sólo el valor de la frecuencia inicial, y las sucesivas notas tendrán la frecuencia de la anterior multiplicada por 1,059. Usamos Programación Multitarea.

GUIA 04 – (Usa Lista de Materiales 02)

-	LIBRERIAS CON ARDUINO (Parte 1)
1	Nuestra primer Librería. Este programa envía por “Monitor Serie”, el resultado de sumar dos números. La suma es realizada por una función de la librería llamada “Calculista.h”, creada por nosotros (Algo muy simple para comenzar).
2	Trabajando con Librerías Propias 01. En este ejercicio, trabajaremos nuevamente con el programa “004_Parlante_03a” Que usamos en la “guía 3” y el enunciado dice: “Emitir Beep-Beep por el Speaker/Zumbador- (Versión B) – Programación Multitarea.
3	Trabajando con Librerías Propias 02. Acá les dejo unas melodías de películas: Harry Potter y Star Wars. El objetivo es usar librerías creadas por nosotros.
4	PROGRAMA QUE CONTIENE BIBLIOTECA DE SONIDOS MUY DIFUNDIDA, PRÁCTICAMENTE SE PUEDE CONSIDERAR STANDARD.
5	LIBRERIAS CON ARDUINO (Parte 2) – Instalación Manual.
6	LIBRERIAS CON ARDUINO (Parte 3) – Instalación con el IDE.
7	Visualizar Ejemplos Que Contienen Las Librerías.

Anexo de Programación I (Reconocimiento de Texto)

-	EN ESTE ANEXO, SE AGRUPAN, EXPLICAN Y EJEMPLIFICA LAS FUNCIONES DE RECONOCIMIENTO Y PROCESAMIENTO DE TEXTO. A continuación veremos una serie de funciones, que comenzaremos a usar en el momento que enviemos órdenes a nuestro dispositivo, ya sea comandos enviados por el puerto serie, leyendo un archivo con ordenes pregrabadas (Secuencia de comandos) y/o archivos de inicialización, etc.
---	---

GUIA 05 – (Usa Lista de Materiales 03)

1	Reconocimiento de una Foto Resistencia LDR - Lectura de Valores Entregados. Lecturas de los valores entregados por una Foto Resistencia LDR. Visualización de Resultados en el monitor del Puerto Serie.
2	Busca e Informa la Mayor y Menor intensidad de luz Registrada por la Fotorresistencia LDR. Buscar y visualizar la mayor y menor luminosidad registrada por la Fotorresistencia LDR a través del monitor del Puerto Serie.
3	Prende luz nocturna automáticamente. Al anochecer, disminuye la luz y este circuito, prendera una luz Nocturna, luego al amanecer, aumenta la luminosidad, y la luz se apagara automáticamente. La intensidad de la luz luminosidad se registrará con una Fotorresistencia LDR.
4	Prende luz nocturna automáticamente (Permite Regular que tan de noche se prenderá). Al anochecer, disminuye la luz y este circuito, prendera una luz Nocturna

	(Permitiendo regular mínimo de luz natural para prender con un Potenciómetro) , luego al amanecer, aumenta la luminosidad, y la luz se apagará automáticamente. La intensidad de la luz (luminosidad) se registrará con una Fotorresistencia LDR
5	Informa la intensidad de luz Registrada por la Fotorresistencia LDR visualmente con 6 LEDs. Informar la luminosidad registrada por la Fotorresistencia LDR a través de 6 leds (Vúmetro Lumínico) y del monitor del Puerto Serie. (Un vúmetro no es más que un instrumento el cual nos indica el nivel de la señal que nos entrega el sensor, y nosotros lo que haremos será representarlo con unos leds para que sea mas vistoso).
6	Escuchemos cuanta Luz Detecta una Fotorresistencia LDR. Generar sonido proporcionalmente a la luz registrada por el sensor. Paralelamente, accionar un LED, para que se prenda acompañando el sonido, siempre acorde a la cantidad de luz registrada.
7	Reconocimiento de un Sensor de Distancia Ultrasónico HC-SR04. Cálculo de distancia con Valores Entregados. Lecturas de los valores entregados y cálculo de la distancia. Visualización de Resultados en el monitor del Puerto Serie.
8	Alarma de Proximidad – Versión 01. Prender un Led rojo como Alarma, cuando un objeto se acerque más de una distancia de seguridad. Muestra por el puerto serie la distancia calculada.
9	Alarma de Proximidad - Versión 02. Alarma de Proximidad Con Sonido y Luces – Área de seguridad (configurable con potenciómetro). Muestra por el puerto serie la distancia calculada/Configurada.
-	Ideas para mejorar Alarma.

GUIA 06 – (Usa Lista de Materiales 03)

1	Reconocimiento de un Servo Motor. Proyecto A - Usando For. Hacemos girar en toda su amplitud a un servo motor (180). Para este ejemplo usamos el Servo Motor MG90S. Debido a su bajo consumo y gran fuerza.
2	Reconocimiento de un Servo Motor. Proyecto B – Programación Multi Tarea (Este es la forma de programar Recomendada). Hacemos girar en toda su amplitud a un servo motor (180). Para este ejemplo usamos el Servo Motor MG90S. Debido a su bajo consumo y gran fuerza.
3	Servo Motor controlado por un Potenciómetro. Con un potenciómetro, controlar el ángulo de giro de un Servo Motor. Al Angulo de giro debo oscilar entre 0 y 180.
4	Hacer Girar Servo Motor controlado por un dos Botones (Izquierdo y derecho). Prever límite izquierdo y derecho del servo y usar una alarma. Proyecto A. Con dos botones hacer girar un Servo Motor, avisando con Una alarma de luz y sonido cuando se presione ambos botones al mismo tiempo, y alarma Izquierda o derecha cuando se intente hacer girar mas allá del imite al servo. Recordar que un Servo Motor tiene un Angulo de giro que oscilar entre 0° y 180°. Este proyecto lo haremos dos veces , en este, el proyecto “A”, mientras mantenemos apretado un botón, el servo gira hasta llegar al tope (este resulta muy practico pero muy inexacto al momento de necesitar un giro preciso).
5	Hacer Girar Servo Motor controlado por un dos Botones (Izquierdo y derecho). Prever límite izquierdo y derecho del servo y usar una alarma. Proyecto B. Con dos botones hacer girar un Servo Motor, avisando con Una alarma de luz y sonido cuando se presione ambos botones al mismo tiempo, y alarma Izquierda o derecha cuando se intente hacer girar mas allá del imite al servo. Recordar que un Servo Motor tiene un Angulo de giro que oscilar entre 0° y 180°. En este Proyecto, el B , cada vez que presionamos un botón realiza un pequeño giro (valor de cada salto es configurable) que resulta muy preciso, a la hora de hacer un pequeño giro.
6	Seguidor de Luz (Girasol) simple - Busca a la derecha o a la izquierda – usa un

	único servo. Proyecto A. Este Dispositivo tiene la capacidad de buscar el ángulo, en el que recibe la mayor cantidad de luz, sin importar si proviene de un reflejo, o directamente del sol. Tiene una gran desventaja, y es que no puede girar en todas las direcciones
7	Radar Ultrasónico, posee una capacidad de giro de 180 grados, usa un único Servo Motor y un sensor Ultrasónico. Proyecto A. Este dispositivo es capaz de detectar un objeto e indicar a que distancia se encuentra. Solo esta limitado por la capacidad de giro del servo motor, y por la distancia que nuestra servo es capaz de soportar.
8	Radar Ultrasónico – Posee una capacidad angular para detectar de 320 grados girando solo 180°. Proyecto B. Para esto usaremos 2 sensores Ultrasónicos, rotados como se muestra en la figura y un Servo Motor. Este dispositivo es capaz de detectar un objeto e indicar a que distancia se encuentra. Solo esta limitado por la capacidad de giro del servo motor, y la distancia que nuestros sensores sean capaces de controlar.
9	Manejar un Servo desde la PC por el puerto Serie. Proyecto A. El programa leerá un número entero (Angulo de rotación) por el puerto serie y posicionará el servo.
-	Apéndice A - Librería “Servo.h” Clases y Métodos disponibles.

GUIA 07 – (Usa Lista de Materiales 04)

1	Movimientos básicos de un Servo Motor con alimentación externa. La placa Arduino continuará siendo alimentada desde la PC por el cable USB, mientras controla el Servo Motor, que será alimentado con 4 Pilas comunes AA (1,5V - No Recargables).
2	Servo Motor controlado por un Potenciómetro con alimentación externa. La placa Arduino continuará siendo alimentada desde la PC por el cable USB, mientras controla el Servo Motor y valores entregados por el Potenciómetro, quines serán alimentados con 4 Pilas comunes AA (1,5V - No Recargables).
3	Radar Ultrasónico con alimentación externa. – Posee una capacidad angular para detectar de 320 grados girando solo 180°. Este dispositivo es capaz de detectar un objeto e indicar a que distancia se encuentra. La placa Arduino continuará siendo alimentada desde la PC por el cable USB mientras controla el servo motor, los Led y los 2 Sensores de Distancia Ultrasónico HC-SR04, que serán alimentados con 4 Pilas comunes AA (1,5V - No Recargables).
4	Hacer Girar Servo Motor controlado por un dos Botones (Izquierdo y derecho) y alimentación externa - Prever límite izquierdo y derecho del servo y usar una alarma. Con dos botones hacer girar un Servo Motor, avisando con Una alarma de luz y sonido cuando se presione ambos botones al mismo tiempo, y alarma Izquierda o derecha cuando se intente hacer girar mas allá del imite al servo. La placa Arduino continuará siendo alimentada desde la PC por el cable USB mientras controla los 3 Led, el Speaker, los 2 botones y el servo motor, que será alimentados con 4 Pilas comunes AA (1,5V - No Recargables).
5	Buscador de Luz o Girasol - Proyecto B. - Este dispositivo, usando alimentación externa, es capaz de girar y buscar la luz en cualquiera de los cuatro puntos cardinales, se vale de dos Servo Motores y cuatro Fotorresistencias LDR.
-	Conexión de 8/16 servos con alimentación Externa, controlados por Arduino.
6	Placa Controladora PCA9685 - Mover 4 SevoMotores de los 16 posible que nos permite mover. Solo a modo de ejemplo, se toman 4 servos y se las da diferentes posiciones. Este proyecto esta pensado para tener alimentación de una Fuente de 5V 5A.
7	Mover 4 SevoMotores (distintos) de los 16 posible que nos permite mover la Placa Controladora PCA9685. En el caso que usen diferentes tipos de servomotores con diferente rango del ancho de pulso.
-	Apéndice A - Librería “Adafruit_PWMServoDriver.h” - Clases y Métodos

	disponibles.
-	Apéndice B – Direccionamiento de una Placa Controladora PCA9685.

GUIA 08 – (Usa Lista de Materiales 04)

1	Grabar un Archivos Con 10 Valores tomados de un Sensor Ficticio. Usando una Lectora/Grabadora de Tarjetas SD, generar un Archivo con 10 lecturas de un censored ficticio. La función que simula leer el sensor solo retorna cero. Cada línea grabada contendrá, el número de línea, el tiempo (expresado en milésimas de segundo), desde que se activo el programa, y el valor del sensor, en nuestro programa será cero. El nombre del archivo es "datalog.txt"
2	Leer el Archivo generado anteriormente y mostrar los datos por el monitor del puerto Serie. Leer el archivo "datalog.txt" que generamos en el programa anterior. Mostrar los datos leídos en la PC, por el Monitor del Puerto Serie.
3	Programa que muestra por monitor del Puerto Serie, el estado y características de la Tarjeta. También mostrar información de los archivos contenidos. Este ejemplo es para quienes le interesa profundizar en medios de almacenamientos. Datos muy importantes desde el comienzo de la informática, y que actualmente el usuario común desconoce.
4	Grabar en archivo los valores entregados por 4 Potenciómetros. Usar Alimentación Externa para la Lecto/Grabadora, y Una variable que contenga el Nombre de Archivo. En este ejercicio guardaremos en un Archivo los valores arrojados por 4 potenciómetros y mientras aprenderemos como poner el nombre del Archivo dentro de una variable, esto es muy importante para futuros proyectos.
5	Leer el archivo generado con los Valores entregados por 4 Potenciómetros y mostrarlo por el monitor del Puerto Serie. Usar Alimentación Externa para la Lecto/Grabadora, y Una variable que contenga el Nombre de Archivo. En este ejercicio recuperamos (leemos) de un Archivo, los valores (Números enteros separados por coma) grabados en el ejemplo anterior. Usamos el nombre del Archivo dentro de una variable, esto es muy importante para futuros proyectos.
6	Verificar La Existencia de un Archivo, luego intentar Crearlo y finalmente intentar Borrarlo. Emitiendo en todos los casos los mensajes que corresponden. La finalidad de este ejemplo es mostrar la simplicidad con la que podemos trabajar archivos, que luego nos resultara muy útil a la hora de manejar dispositivos complejos.
7	Creación y Uso de Carpetas (Directorios) dentro de la Tarjeta SD. Manejo de Archivos y listado completo de Archivos y carpetas. En este Proyecto, será realizado en tres etapas (A, B, C), la primera, creamos el ambiente de trabajo, además de guiarlos en la creación y borrado de Carpetas (Directorios) y profundizamos en la creación de Archivos. En la segunda etapa, jugaremos un poco en el recorrido de la estructura de la Tarjeta (carpetas). Y finalmente en la tercera nos adentramos en algoritmos recursivos, algo prácticamente indispensable para recorrer completamente nuestra tarjeta.
-	Apéndice A - Librería "SD.h" Clases y Métodos disponibles.
-	Apéndice B - Ventana de Comandos o Símbolo del Sistema.
-	Apéndice C – Recursividad.

GUIA 09 – (Usa Lista de Materiales 05)

-	DESCRIPCION Motor Robot DC 3-6v + Caja reductora
-	DRIVER L298N - Para controlar motores DC y paso a paso - Todo lo que hay que saber. Guía de conexión.
-	<u>Donde ENCUENTRO MAS JUMPER para Configurar mi DRIVER?</u>

1	Primeros Movimientos de un Motor DC de 6v – Avance, Retroceso y Parar – Usamos controlador L298N. Este programa nos permite controlar los movimientos básicos de un motor DC de 6V. Adelante, Atrás y Parar. Tenemos activa la Capacidad de Controlar la velocidad, pero la dejamos fija en 150 (recordar que la velocidad puede oscilar entre 0 y 255) - En este caso usamos el Motor Izquierdo.
2	Manejar dos Motores DC de 6v simultáneamente, controlando Avance, Retroceso, Parar, Giro a la derecha y giro a la Izquierda – Usamos controlador L298N. Este programa nos permite controlar los movimientos de un motor DC de 6V. Tenemos activa la Capacidad de Controlar la velocidad, pero la dejamos fija en 150 (recordar que la velocidad puede oscilar entre 0 y 255).
3	Manejar dos Motores DC de 6v simultáneamente, controlando Avance, Retroceso, Parar, Giros a la derecha e Izquierda – Usamos controlador L298N y Librería "LEANTEC_ControlMotor". Este programa nos permite controlar los movimientos de un motor DC de 6V. Tenemos activa la Capacidad de Controlar la velocidad (recordar que la velocidad puede oscilar entre 0 y 255). En este proyecto ganamos la capacidad de controlar cuanto queremos girar.
4	Controlar el Avance y Retroceso de dos Motores DC de 6v simultáneamente por medio de un Potenciómetro – Usamos controlador L298N y Librería "LEANTEC_ControlMotor". Este programa nos permite controlar los movimientos adelante y atrás de dos motores DC de 6V. Tenemos activa la Capacidad de Controlar la velocidad (recordar que la velocidad puede oscilar entre 0 y 255).
	Control desde la PC
	Ordenes pre grabadas en archivo
	Al fin un dispositivo se desplaza sin cables.! - Alimentación de Arduino con Pilas -
-	<u>Apéndice A - Puente H</u>

GUIA 10 – (Usa Lista de Materiales 05)

1	Enviar un número por Bluetooth y Arduino hará parpadear un Led conectado a la placa, el número de veces que indique el número enviado (entre 1 y 9). Enviando comandos desde nuestro celular, Arduino lo recibe y Hacemos parpadear un LED. El comando enviado desde el celular es un número que deberá estar previamente configurado: Cuando apretamos una tecla en el celular, se enviará el número.
2	Mostrar en el monitor del puerto serie, los caracteres (comandos) que ingresan por el modulo Bluetooth. Con este programa podremos analizar lo que estamos leyendo del modulo Bluetooth, cada vez que necesitemos comprender. Luego podremos programar el que hacer ante tal o cual comando (Ver Apéndice B - Librería SoftwareSerial). El circuito es el mismo para ambos programas.
3	Enviar un número por Bluetooth y Arduino hará parpadear un Led conectado a la placa, el número de veces que indique el número enviado (entre 1 y 9). La única diferencia con el primer ejemplo, es que gracias a la librería SoftwareSerial nos podemos comunicar con la PC y con el modulo Bluetooth simultáneamente. (Ver Apéndice B - Librería SoftwareSerial).
4	Comandos: Prender y/o apagar un Led, enviando el comando através de Bluetooth: 1 Prende y 0 Apaga. Esta forma de trabajar permitirá reconocer infinidad de comando y realizar acciones asociadas a los comandos, manteniendo comunicación con el monitor del puerto serie simultáneamente.
5	Mostrar en el monitor del puerto serie, las cadenas de caracteres o palabras (comandos) que ingresan por el modulo Bluetooth. Con este programa podremos

	analizar lo que estamos leyendo del modulo Bluetooth, cada vez que necesitemos comprender. Luego podremos programar el que hacer ante tal o cual comando (Ver Apéndice B - Librería SoftwareSerial). El circuito es el mismo para ambos programas.
6	Mostrar en el monitor del puerto serie, las cadenas de caracteres o palabras (comandos) que ingresan por el modulo Bluetooth. Incorporar una función de ayuda al usuario. Con este programa podremos analizar lo que estamos leyendo del modulo Bluetooth, cada vez que necesitemos comprender. Luego podremos programar el que hacer ante tal o cual comando (Ver Apéndice B - Librería SoftwareSerial). El circuito es el mismo para ambos programas.
7	Controlar un Servo Motor por medio de comandos recibidos por Bluetooth. Enviar comandos desde Celular por Bluetooth, indicar el Angulo en el que debe posicionarse el Servo Motor. Los valores (ángulos) permitidos son todos los mayores o iguales a cero y menores o iguales a 180.
8	Controlar un Vehículo (dos motores) por medio de comandos recibidos por Bluetooth. Este programa permite recibir comando y ejecutar las funciones básicas en un automóvil (A- adelante, R- atrás, D- derecha, I- izquierda, S- alto). El código es muy similar al que usamos para controlar un automóvil desde la PC.
-	Configuración Modulo Bluetooth - Comandos AT - (Modelos HC-05 y HC-06)
-	Apéndice A - Comandos Bluetooth desde un Celular. Este Apéndice incluye comandos de texto y Comando de Voz.
-	Apéndice B - Librería SoftwareSerial. (SoftwareSerial.h)

GUIA 11 – (Usa Lista de Materiales 05)

1	Dispositivos controlados desde un Control remoto
2	

GUIA 12 – (Usa Lista de Materiales 06)

1	Memoria EEPROM + Archivos de Inicialización

GUIA 13 – (Usa Lista de Materiales 07)

1	Motores Paso a Paso

GUIA 14 – (Usa Lista de Materiales 07)

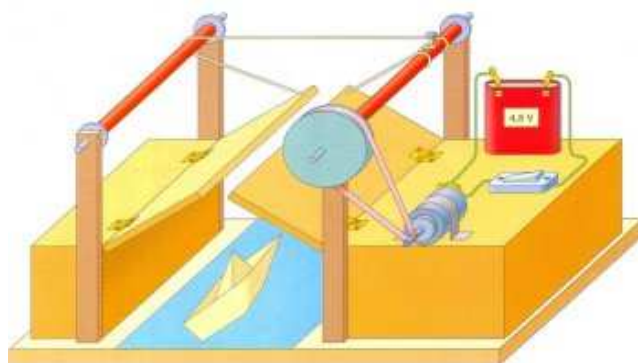
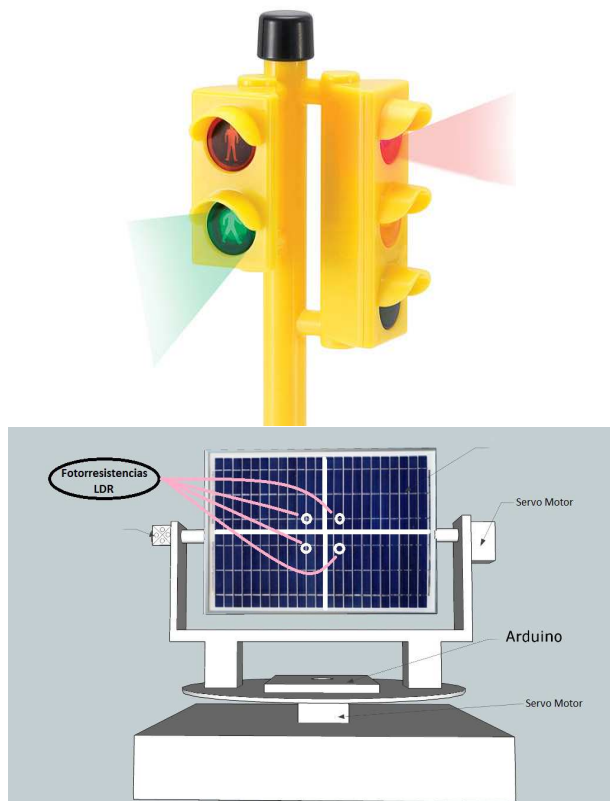
1	Reconocimiento de un Sensor de Temperatura LM35. Lectura de Valores Entregados. Lecturas de los valores entregados por un Sensor de Temperatura LM35. Visualización de Resultados en el monitor del Puerto Serie.

IDEAS – (Usa Lista de Materiales XX)

	Modulo Sensor Presión Atmosférica
	Sensor De Sonido con Micrófono Regulable - Aplausómetro y otros
	Sensor De Obstáculos Infrarrojo
	Prender Apagar uno o varios Cooler (Ventiladores)
	Display 1 y 2 Dígitos Numéricos
	Led Laser Rojo de 5V de 5mW
	Compás Brújula Electrónica

Otras Actividades Desarrolladas

	Uso de Relé
	Sensor Óptico Reflectivo Infrarrojo
	Módulos Rf Emisor Y Receptor)
	Sensor Touch (Táctil)
	Tarea en paralelo (ejemplo monitorear Temperaturas y Accionar Cooler)
	Estudio de estructuras de la Programación, ciclo “while”
	Estudio de estructuras de la Programación, ciclo “do while”
	Estudio de estructuras de la Programación, Vectores.
	Estudio de estructuras de la Programación, Matrices
	Y Más...





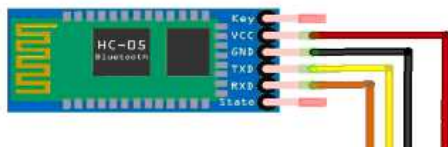
CONTROL REMOTO INFRARROJO



Imagina..!



bluetooth voice command



Reconocimiento de Voz

Si tienes algunas Correcciones y/o Sugerencias, por favor contáctame.

El Sitio Oficial de Arduino es:

www.arduino.cc



```
01000101 01110011 01100011 01110101 01100011 01101000 01101111 00100000 01100001
00100000 01101100 01101111 01110011 00100000 01100010 01100101 01100010 11000011
10101001 01110011 00100000 01101100 01101100 01101111 01110010 01100001 01101110
01100100 01101111 00101100 00100000 01101100 01101111 01110011 00100000 01110110
01100101 01101111 00100000 01100011 01110010 01100101 01100011 01100101 01110010
00101110 01000101 01110011 01110100 11000011 10100001 01101110 00100000 01100001
01110000 01110010 01100101 01101110 01100100 01101001 01100101 01101110 01100100
01101111 00100000 01101101 11000011 10100001 01110011 00100000 01100100 01100101
00100000 01101100 01101111 00100000 01110001 01110101 01100101 00100000 01111001
01101111 00100000 01101110 01110101 01101110 01100011 01100001 00100000 01010011
01100001 01100010 01110010 01100101 00101110 01011001 00100000 01101101 01100101
00100000 01100100 01101001 01100111 01101111 00100000 01100001 00100000 01101101
01101001 00100000 01101101 01101001 01110011 01101101 01101111 00101100 00100000
01110001 01110101 01100101 00100000 01101101 01110101 01101110 01100100 01101111
00100000 01110100 01100001 01101110 00100000 01101101 01100001 01110010 01100001
01110110 01101001 01101100 01101100 01101111 01110011 01101111 00101110 01010011
11000011 10101101 00101100 00100000 01101101 01100101 00100000 01100100 01101001
01100111 01101111 00100000 01100001 00100000 01101101 01101001 00100000 01101101
01101001 01110011 01101101 01101111 00101100 00100000 01110001 01110101 01100101
00100000 01101101 01110101 01101110 01100100 01101111 00100000 01110100 01100001
01101110 00100000 01101101 01100001 01110010 01100001 01110110 01101001 01101100
01101100 01101111 01110011 01101111 00101110
```

Si tienes algunas Correcciones y/o Sugerencia, por favor contáctame.