Continue

50%(2)50% encontró este documento útil (2 votos)170 vistas91 páginasInformación del documentoGuardarGuardar Libro-Arduino-Cookbook-2nd-Edition.pdf para más tarde50%50% encontró este documento útil, undefined Aquí os paso un link con una recopilación de libros para quien no tenga dinero para comprar los libros o no puede obtenerlos por lo que sea. En principio descargarlo si no lo tenéis, pero si os gusta y queréis apoyar al autor, compradlos por favor Los libros que encontraréis serán: Minibloq + Arduino. Utilización del entorno de programación Minibloq para programar la Tarjeta Arduino Ardublock + Arduino. Utilización de Ardublock para la generación en modo gráfico de aplicaciones para el IDE Arduino S4A Scratch + Arduino. Utilización de S4A (Scratch) más la tarjeta Arduino en un ambiente de programación gráfica orientado a la educación MyOpenlab + Arduino. Una propuesta de Utilización de Open Hardware y Software Libre GNU para el Diseño y Simulación de Prototipos en el Laboratorio Entornos Gráficos para la Programación: Plataforma Open Hardware Arduino Guía de Usuario de Arduino Arduino programing notebook de Brian Evans Ejercicios de Arduino resueltos Introducción a Arduino Link:otra firma - Total Mangas Anime & Photoshop +5913 1 Like El enlace que has puesto ya lo han borrado, tienes otro nuevo? Hay posibilidad de enviarmelo por mp? Buenas, estoy en modo aprendizaje y buscando encontré esto por si te vale. Taringa! – 8 Feb 12 Martin95 : Minibloq + Arduino. Utilización del entorno de programación Minibloq para programar la Tarjeta Arduino(66pp / 2,76MB) Con esta quinta... : Ardui 0 calificaciones0% encontró este documento útil (0 votos)153 vistas19 páginasInformación del documentoGuardarGuardar Arduino Cookbook 2nd Edition 1 para más tarde0%0% encontró este documento útil, undefined Gracias al estupendo trabajo de Rafael Enriquez Salvador, ahora ya está disponible para su descarga el manual completo del libro "Arduino Programming Notebook" totalmente en Español. Lo ha publicado en el Laboratorio de Software Libre de la Universidad de Córdoba bajo una licencia Creative Commons Share Alike 2.5. También facilita la descarga del código fuente de ejemplo del libro. Éste documento es imprescindible para todos los que empiecen o quieran comenzar con la plataforma Arduino. Estupenda aportación!! Enlace a la descarga a continuación... Enlaces: Manual de usuario Arduino en Español y código fuente Si te ha gustado, compártelo con tus amigos! Facebook Twitter Ver comentarios antiguos... Guia para programar como un profesional Solo por suscribirte, recibirás una guía gratuita en PDF sobre cómo instalar y configurar PlaformIO para llevar tus proyectos al siguiente nivel APUNTATE AQUI Recetas para iniciar, expandir, y mejorar sus proyectos EN ESPAÑOL TRADUCCIÓN: HENRY ALVARADO Prólogo vii Electrónica/Software Ingeniería/Robótica Arduino Cookbook • Usar Arduino con una variedad de populares dispositivos de entrada y sensores “El conjunto de recetas de Michael es un buen regalo para la creciente comunidad de Arduino. Cualquiera que sea su conocimiento o habilidad, el libro provee soluciones para ese proyecto con el que está luchando hoy en día y estímulo para imaginar lo que construirá mañana. Dudo que alguna vez dejaré mi mesa de trabajo” • Controlar pantallas visuales, generar sonido, y controlar varios tipos de motores -Mikal Hart Arduino Uno Asesor Equipo ¿Desea crear dispositivos que interactúen con el mundo físico? Este libro es perfecto para cualquiera quien desee experimentar con el popular microcontrolador Arduino y su entorno de programación. Usted encontrará más de 200 consejos y técnicas para construir una variedad de objetos y prototipos como juguetes, detectores, robots y ropa interactiva que puede sentir y responder al tacto, sonido, posición, calor, y luz. Usted no necesita experiencia con Arduino o programación para iniciar. Actualizado para Arduino 1.0 liberado, las recetas en esta segunda edición incluyen ejemplos prácticos y orientación para ayudarle a empezar, expandir, y mejorar sus proyectos inmediatamente ya sea usted un artista, diseñador, aficionado, estudiante o ingeniero. • Ponerse al día rápidamente sobre la placa Arduino y conceptos esenciales de software • Aprender técnicas básicas para lectura digital y señales analógicas • Interactuar con dispositivos que utilizan controles remotos, incluyendo Televisores y electrodomésticos • Aprender técnicas para manejo retrasos de tiempo y medición del tiempo • Aplicar avanzada codificación y manejo de técnicas de memoria Michael Margolis es un tecnólogo en el campo de la computación en tiempo real, con experiencia en el desarrollo de hardware y software para interactuar con el entorno. Él tiene más de 30 años de experiencia en niveles superiores con Sony, Microsoft, and Lucent/Bell Labs, y ha escrito librerías y núcleo de software incluido en la distribución Arduino 1.0 Prólogo viii SEGUNDA EDICIÓN Arduino Cookbook Michael Margolis Arduino Cookbook, Segunda Edición Michael Margolis Derechos de autor © 2012 Michael Margolis, Nicholas Weldin. Todo derechos reservados Impreso en los EEUU Publicado por O'Reilly Medios, Inc., 1005 Gravenstein Carretera Norte, Sebastopol, California 95472. O'Reilly libros puede ser comprado para educativa, negocio, o venta promocional utilizar. Online ediciones son también disponible para más títulos ( ). Para más información, póngase en contacto nuestro corporativo/institucional departamento de venta: (800) 998-9938 o . Editores: Shawn Wallace y Brian Jepson Producción Editor: Teresa Elsey Corrector de pruebas: Kiel Furgoneta Cuerno Indexador: Lucie Haskins Cubierta Diseñador: Karen Montgomery Interior Diseñador: David Futato Illustrator: Robert Romano Marzo 2011: Primero Edition.Diciembre 2011: Segundo Edition. Revisión Historia para la Segundo Edición: 2011-12-09 Primero liberación Ver para más detalles. Nutshell Handbook, el logotipo de Nutshell Handbook, y el logotipo de O'Reilly son marcas registradas de O'Reilly Media, Inc. Arduino Cookbook, la imagen de un juguete conejo, y conexo comercio vestido son marcas comerciales de O'Reilly Medios, Inc. Muchas de las designaciones usadas por los fabricantes y los vendedores para distinguir su productos son reclamados como marcas. Donde aquellas designaciones aparecen en este libro, y O'Reilly Medios, Inc., era consciente de una marca comercial reclamar, la designaciones tener estado impreso en tapas o inicial gorras. Mientras cada precaución tiene han tomado en la preparación de este libro, la editor y autores asumirá ninguna responsabilidad para errores o omisiones, o para daños y perjuicios resultante desde la utilizar de la información mantenida con- en el presente documento. ISBN: 978-1-449-31387-6 [LSI] 1323465788 Prólogo Este libro fue escrito por Michael Margolis con Nick Weldin para ayudarle a usted explorar las increíbles cosas usted puede hacer con Arduino. Arduino es un familia de microcontroladores (Pequeños ordenadores) y un entorno de creación de software que hace fácil para usted crear programas (Llamados sketches) que pueden interactuar con el mundo físico. Las cosas que usted puede hacer con Arduino pueden sentir y responder al tacto, sonido, posición, calor, y luz. Este tipo de tecnología, a menudo referido a como computación física, es usado en todo tipo de cosas desde el iPhone hasta sistemas electrónicos de automóviles. Arduino hace esto posible para cualquier persona con un interés, incluso personas sin experiencia en programación o electrónica para usar esta rica y compleja tecnología. ¿A quién va dirigido este libro? A diferencia de otros libros, la experiencia con software y hardware es no indispensable. Este libro es dirigido a los lectores interesados en el uso de la tecnología de computación para interactuar con el entorno. Es para personas que quieren a rápidamente encontrar la solución a los problemas de hardware y software. Las recetas proporcionan la información que usted necesita para llevar a cabo un amplio alcance de tareas. También tiene detalles para ayudarle a personalizar soluciones para cumplir sus necesidades específicas. Hay insuficiente espacio en un libro limitado a 700 páginas cubrir a fondo la teoría general, así que hay Enlaces a referencias externas son proporcionados a lo largo de la libro. Ver "Qué Fueron dejados en la página xiv” para algunas referencias generales para aquellos sin experiencia en programación o electrónica. Si usted no tiene experiencia en programación experiencia, tal vez usted tiene un gran idea para un proyecto interactivo pero no tiene la habilidades para desarrollarlo, este libro le ayudará a aprender lo que necesita saber para escribir código que funcione, usando ejemplos que cubran más de 200 tareas comunes. Si usted tiene alguna experiencia en programación pero es nuevo en Arduino, el libro le ayudará a ser productivo rápidamente demostrando cómo implementar específicamente las capacidades de Arduino para su proyecto. Prólogo xi Personas que ya han usado Arduino deberían encontrar el contenido útil para rápido aprendizaje de nuevas técnicas, que son explicadas usando ejemplos prácticos. Este le ayudará a embarcarse en proyectos más complejos mostrando cómo resolver problemas y utilizando capacidades que pueden ser nuevas para usted. Experimentados programadores en C / C + + pueden encontrar ejemplos de cómo a utilizar recursos de bajo nivel AVR (interruptores, temporizadores, I2C, Ethernet, etc) para construir aplicaciones que utilizan el entorno Arduino. Estructura del libro El libro contiene información que cubre el amplio alcance de las capacidades de Arduino, desde básicos conceptos y tareas comunes, hasta tecnología avanzada. Cada técnica es explicada en una receta que muestra cómo implementar una capacidad específica. Usted no necesitar a leer el contenido en secuencia. Donde una receta utiliza una técnica cubierta en otra receta, el contenido en la otra receta hace referencia en lugar de repetir detalles en múltiples lugares. Capítulo 1, Empezando, introducción al entorno de Arduino y proporcionar ayuda en conseguir el entorno de desarrollo Arduino y hardware instalado y trabajando. El próximo par de capítulos introduce al desarrollo de software Arduino. Capítulo 2, Haciendo al Sketch realizar su orden, cubre conceptos y tareas esenciales de software, y el Capítulo 3, Usando operadores Matemáticos, muestra cómo a hacer uso de las más comunes funciones matemáticas. Capítulo 4, Comunicaciones Serie, describe cómo a consigue Arduino a conectarse y comunicarse con su ordenador y otro dispositivos. Puerto Serie es el más común método para entrada y salida de Arduino, y esta capacidad es usada en muchas de las recetas de todo el libro. Capítulo 5, Entrada Digital y Analógica simple, introduce un alcance de básico técnicas para la lectura de señales digitales y analógicas. Capítulo 6, Entradas desde Sensores, construye sobre esto con recetas que explican cómo a utilizar dispositivos que permiten a Arduino sentir el tacto, sonido, posición, calor, y luz. Capítulo 7, Salidas Visuales, acerca de controladores de luz. Recetas sobre conmutación en uno o muchos LEDs y control de brillo y de color. Este capítulo explica cómo usted puede controlar barras gráficas y displays LED numéricos, así como crear patrones y animaciones con matrices de LED. Además, el capítulo proporciona una introducción general a salidas digitales y analógicas para aquellos que son nuevos en esto. Capítulo 8, Salida física, explica cómo usted puede hacer mover cosas controlando motores con Arduino. El amplio alcance de tipos de motores es cubierta: solenoides, servo motores, DC motores, y motores paso. Prólogo xii Capítulo 9, Salida de Audio, muestra cómo a generar sonido con Arduino a través de un dispositivo de salida como un altavoz. Esto cubre reproducción simple de tonos y melodías y reproducción de archivos WAV y MIDI. Capítulo 10, Controlando Remotamente Dispositivos Externos, describe técnicas que pueden utilizarse para interactuar con casi cualquier dispositivo que usa alguna forma de control remoto, incluyendo TV, equipos de audio, cámaras, puertas de garaje, electrodomésticos, y juguetes. Esto se basa en técnicas usadas en anteriores capítulos para conectar Arduino a dispositivos y módulos. Capítulo 11, El uso de pantallas, acerca de interfaz de LCD de texto y gráficos. El capítulo muestra cómo usted puede conectar estos dispositivos para visualización de texto, desplazar o resaltar palabras, y crear símbolos y caracteres especiales. Capítulo 12, Uso de Tiempo y Fechas, cubre la construcción en Arduino relacionado a funciones de tiempo e introduce muchas técnicas adicionales para la manipulación de tiempo de retrasos, medición de tiempo, y tiempos y fechas en el mundo real. Capítulo 13, Comunicación mediante I2C y SPI, acerca de los estándares, Circuito Inter-Integrado (I2C) e Interfaz Serie Periférica (SPI). Estos estándares proporcionan formas sencillas para información digital a ser transferido entre sensores y Arduino. Este capítulo muestra cómo a utilizar I2C y SPI para conectar a dispositivos comunes. Esto también muestra cómo para conectar dos o más placas Arduino, usando I2C para aplicaciones multi-placas. Capítulo 14, Comunicación Inalámbrica, Acerca de la comunicación inalámbrica con XBee y otros módulos inalámbricos. Este capítulo proporciona ejemplos que van desde simple puerto serie de reemplazo inalámbrico a redes de malla conectando múltiples placas a múltiples sensores. Capítulo 15, Ethernet y Networking, describe las muchas formas que usted puede utilizar Arduino con la Internet. Este tiene ejemplos que demuestran cómo a construir y utilizar clientes y servidores web y muestra cómo a utilizar la más común conexión comunicación a protocolos de internet con Arduino. Las librerías de software Arduino son una forma estándar de añadir funcionalidad al Entorno de Arduino. Capítulo 16, El uso, Modificación, y Creación Librerías, explica cómo utilizar y modificar librerías de software. Este también proporciona dirección en cómo a crear sus propias librerías. Capítulo 17, Codificación Avanzada y Manipulación de Memoria, cubre técnicas avanzadas de programación, y la temas aquí son más técnicos que las otras recetas en este libro porque ellos cubren cosas que son generalmente ocultas por la amigable envoltura de Arduino. Las técnicas en este capítulo pueden ser usadas para hacer un Sketch más eficiente, ello puede ayudar a mejorar rendimiento y reducir el tamaño del código de sus Sketches. Capítulo 18, Uso del Controlador Chip Hardware, muestra cómo acceder y utilizar funciones de hardware que no son completamente expuestos a través de la documentación de lenguaje Arduino. Esto cubre uso de hardware de bajo nivel de entrada / salida de registros, temporizadores, e interruptores. Prólogo xiii Apéndice A, Componentes Electrónicos, proporciona una visión de conjunto de los componentes utilizados a lo largo del libro. Apéndice B, Uso Esquemáticos, Diagramas y Hojas de Datos, explica cómo a utilizar diagramas esquemáticos y hojas de datos. Apéndice C, Construcción y Conexión del Circuito, proporciona un breve introducción a utilizando un protoboard, conectando y uso fuentes de poder y baterías, y utilizando condensadores para desacoplamiento. Apéndice D, Consejos en Solución de problemas de Software, proporciona consejos en arreglo de problemas de compilación y ejecución. Apéndice E, Consejos en Solución de problemas de Hardware, incluye problemas con circuitos electrónicos. Apéndice F, Pines Digitales y Analógicos, proporciona tablas indicando funcionalidad proporcionada por los pines en placas estándar de Arduino. Apéndice G, ASCII y Ampliado Juegos de caracteres, proporciona tablas que muestra caracteres ASCII. Apéndice H, Migración a Arduino 1.0, explica cómo a modificar código escrito para anteriores lanzamientos para ejecutar correctamente con Arduino 1.0. Qué Fue Dejado Fuera No hay lugar en este libro para incluir teoría y práctica electrónica, aunque orientación es proveída para construir circuitos usados en la recetas. Para más detalle, Los lectores pueden querer a remitir a material que es extensamente disponible en la Internet o a libros tal como la siguiente: • • • • Make: Electronics Charles Platt (O'Reilly, búsquelo en oreilly.com) Getting Started in Electronics por Forrest M. Mims III (Master Publishing) Physical Computing por Dan O'Sullivan y Tom Igoe (Cengage) Práctical Electronics forInventors por Paul Scherz (McGraw-Hill) Este libro explica cómo escribir código para lograr tareas específicas, pero es no una introducción a la programación. Relevantes conceptos de programación son brevemente explicados, pero hay insuficiente lugar para incluir detalles. Si usted desea aprender más acerca de la programación, usted puede querer a remitir a la Internet o a uno de la siguiente libros: • Práctical C Programaming por Steve Oualline (O'Reilly; búsquelo en oreilly.com) • A book on C por Al Kelley y Ira Pohl (Addison-Wesley) Prólogo xiv Mi favorito, aunque no realmente un libro para principiantes, es el libro que yo he usado para aprender Programación C: • The C Programming Language por Brian W. Kernighan y Dennis M. Ritchie (Prentice Hall) Estilo de Código (Acerca del Código) El código usado a lo largo de este libro ha sido adaptado para ilustrar claramente el tema tratado en cada receta. Como una consecuencia, algunos atajos comunes de codificación han sido evitados, particularmente en los primeros capítulos. Programadores experimentados en C a menudo utilizan ricas, pero breve expresiones que son eficientes pero pueden ser un poco difícil comprender para principiantes. Por ejemplo, los primeros capítulos incrementan variables usando expresiones explícitas que son fáciles para no programadores, léase: resultado = resultado + 1 / / incrementa el contador Más bien que la siguiente, comúnmente usada por experimentados programadores, que hace lo mismo: resultado + +; / / incrementa el uso del operador de incremento posterior Siéntase libre de sustituir su estilo preferido. Los principiantes deberían estar tranquilos ya que no hay beneficio en rendimiento o en tamaño de código usando la forma breve. Algunas expresiones de programación son tan comunes que ellas son usadas en su forma breve. Por ejemplo, las expresiones de bucle son escritos de la siguiente manera: for (int i = 0; i < 4; i++) Este es equivalente a la siguiente: int i; for (i = 0; i < 4; i = i+1) Ver Capítulo 2 para más detalles en estas y otras expresiones usadas a lo largo del libro. Una buena práctica de programación implica asegurar que valores usados son válidos (Basura entra igual a basura sale) por revisión antes de usarlos en cálculos. Sin embargo, para mantener el código centrado en tema de la receta, muy poca comprobación de errores código ha sido incluida. Plataforma Arduino Notas de la versión Esta edición ha sido actualizada para Arduino 1.0. Todo el código ha sido probado con la última versión Arduino 1.0 (RC2). La descarga del código para esta edición ha sido actualizada en línea si es necesario apoyo a la versión final 1.0, a fin de comprobar el libro del sitio web para conseguir el más reciente código. La descarga contiene un archivo llamado changelog.txt que indicará código que ha sido cambiado desde la edición publicada. Prólogo xv Aunque muchos de los Sketches ejecutará en las más recientes versiones de Arduino, usted necesita cambiar la extensión desde .ino a .pde para cargar el Sketch en un pre-1.0 IDE. Si usted no ha migrado a Arduino 1.0 y tiene una buena razón para seguir con una versión anterior, usted puede utilizar el código ejemplo en la primero edición de este libro (Disponibles en .com/product/9780596802486.do), que ha sido probado con versiones desde la 0018 a 0022. Tenga en cuenta que muchos recetas en la segunda edición han sido mejoradas, así nosotros motivamos a usted a actualizarse a Arduino 1.0. Si usted necesita ayuda para migrar código antiguo, véase el Apéndice H. Hay también un enlace a erratas en ese sitio. Las erratas dan lectores una manera de hacernos saber acerca de los errores tipográficos, errores, y otro problemas con el libro. Las erratas serán visibles en la página de inmediato, y vamos a confirmar ellas después de revisarlas. O'Reilly puede también corregir erratas en futuras impresiones del libro y en Safari, haciendo para el lector una mejor experiencia rápidamente. Si usted tener problemas haciendo funcionar ejemplos, comprobar el archivo changelog.txt en la última descarga de código a ver si el Sketch ha sido actualizado. Si eso no corrige el problema, consulte el Apéndice D, el cual incluye la solución de problemas de software. El foro Arduino es un buen lugar para publicar una pregunta si usted necesitar más Ayuda: . Si le gusta o no le gusta este libro, por todo medios, por favor que la gente lo sepa. Las reseñas de Amazon son una popular manera para compartir su felicidad u otros comentarios. Usted puede También dejar comentarios en el sitio de O'Reilly para este libro libro. Convenciones Usadas en Este Libro Las siguientes convenciones son usadas en este libro: Cursiva Indica nombres de rutas, nombres de archivo, y nombres de programas; direcciones de internet, tal como Dominios y URL; y nuevo artículos donde ellos son definidos. Constante ancho Indica líneas de comando y Opciones que deberían ser mecanografiado textualmente; nombres y palabras clave en programas, incluyendo método nombres, nombres de variable, y nombres de clases; y elemento y etiquetas HTML Constante ancho audaz Indica énfasis en líneas de código en programas Constante ancho cursivo Indica texto que debería ser reemplazado con valores suministrados por el usuario. Este icono significa un consejo, sugerencia, o nota general. Prólogo xvi Este icono indica una advertencia o precaución. Uso de Ejemplos de Código Este libro está aquí para ayudarle a hacer cosas con Arduino. En general, usted puede utilizar el código en este libro en sus programas y documentación. Usted no necesita a contactarnos para pedir permiso a menos que esté reproduciendo una significativa porción del código. Por ejemplo, escribir un programa use varios trozos de código desde este libro no requiere permiso. Vender o distribuir un CD-ROM de ejemplos de este libro requiere permiso. Contestar una pregunta citando este libro y citando ejemplo de código hace no necesita permiso. La incorporación de una significativa cantidad de ejemplo de código de este libro en su del producto documentación necesita permiso. Nosotros apreciamos, pero no requerimos, atribución. Una atribución en general incluye el título, autor, editor, e ISBN. Por ejemplo: "Arduino Cookbook, Segunda Edición, por Michael Margolis con Nick Weldin (O'Reilly). Derechos de autor 2012 Michael Margolis, Nicholas Weldin, 978-14493-1387-6. " Si usted siente su uso de ejemplos de código cae fuera de su uso justo, o el permiso dado aquí, no dude en contactarnos en . Safari ® Libros Online Safari Libros Online es una biblioteca digital que le permite usted fácilmente buscar encima de 7500 referencias a libros de tecnología y creatividad y Vídeos para encontrar la respuestas usted necesitar rápidamente. Con una suscripción, usted puede leer cualquier página y ver cualquier vídeo desde nuestra biblioteca en línea. Leer libros en su teléfono celular y dispositivos móviles. Acceder a nuevos títulos antes que ellos estén disponibles para imprimir y conseguir exclusivo acceso a manuscritos en desarrollo y publicar comentarios para los autores. Copiar y pegar muestras de código, organizar sus favoritos, descargar capítulos, marcar secciones de libros, crear notas, imprimir páginas, y beneficio de toneladas de otras características de ahorro de tiempo. O'Reilly Medios de comunicación tiene subido este libro al servicio Safari Libros Online. Para tener completo acceso digital a este libro y otros en similares temas desde O'Reilly y otros publicadores, regístrese gratis en . Cómo contactarnos Nosotros hemos probado y verificado la información en este libro a lo mejor de nuestra capacidad, pero puede encontrar que características han cambiado (O incluso que hemos hechos unos pocos errores). Prólogo xvii Por favor, háganos saber acerca de cualquier error que usted encuentre, como bien como sus sugerencias para futuras ediciones, por escrito a: O'Reilly Media, Inc. 1005 Gravenstein Carretera Norte Sebastopol, CA 95472 800-998-9938 (En Estados Unidos o Canadá) 707-829-0515 (Internacional / local) 707-829-0104 (Fax) Nosotros tenemos una página web para este libro, donde enumeramos fe de erratas, ejemplos, y cualquier información adicional. Usted puede acceder a esta página en: Para comentarios o preguntas técnicas acerca de este libro, enviar email a: Para más información acerca de nuestros libros, cursos, conferencias, y noticias, ver nuestro sitio web en . Encuéntrenos en Facebook: Síganos en Twitter: Véanos en YouTube: Agradecimientos La contribución de Mella Weldin fue invaluable para la terminación de este libro. Este fue 90 por ciento escrito cuando Mella vino al consejo y sin su habilidad y entusiasmo, seguiría siendo 90 por ciento escrito. Sus manos en la experiencia Arduino se ejecuta talleres para todos niveles de usuarios habilitados para hacer el asesoramiento en este libro práctico para nuestro amplio alcance de lectores. Gracias, Nick, por su conocimiento y genial, naturaleza colaborativa. Simon St. Laurent fue el editor en O'Reilly quien primero expresó interés en este libro. Y en el final, él es el hombre que sacó esto juntos. Su apoyo y estímulo nos mantuvo inspirados como nosotros tamizamos nuestro camino a través de los volúmenes de material necesario a hacer la materia justa. Brian Jepson me ayudo a comenzar con la escritura de este libro. Su vasto conocimiento de cosas Arduino y su preocupación y experiencia para comunicar acerca de la tecnología claro Inglés establece un alto estándar. Él fue una mano rectora para la conformación del libro y fabricación de tecnología fácilmente accesible para lectores. Nosotros también tenemos a Brian gracias por el contenido de XBee en el Capítulo 14. Brian Jepson y Shawn Wallace fueron editores técnicos para esta segunda edición y proporcionaron excelente asesoramiento para mejorar la precisión y claridad del contenido. Prólogo xviii Audrey Doyle trabajó incansablemente para erradicar errores tipográficos y gramáticos en el manuscrito inicial y desenredar algunas de las más complejas expresiones. Philip Lindsay colaboró en contenido para el Capítulo 15 en la primera edición. Adrian McEwen, el principal desarrollador para muchas mejoras de la Ethernet mejoras en la versión 1.0, proporcionó valioso asesoramiento para asegurar este Capítulo reflejó todos los cambios en esa versión. Mikal Hart escribió recetas sobre GPS y software serie. Mikal fue la elección natural para este no sólo porque él escribió las librerías, sino también porque él es un comunicador con fluidez, un Arduino entusiasta, y gran colaborador. Arduino es posible por la creatividad del equipo de desarrollo del núcleo Arduino; Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, y David Mellis. En nombre de todos los usuarios de Arduino, deseo expresar nuestro agradecimiento por sus esfuerzos en fabricar esta fascinante tecnología simple y su generosidad en hacerlo gratis. Agradecimiento especial a Alexandra Deschamps-Sonsino, cuyos Tinker London workshops proporcionan importante comprensión a las necesidades de los usuarios. Gracias también a Peter Knight, que tiene previsto todo tipo de soluciones inteligentes Arduino como bien como la base de un número de recetas en este libro. En favor de todo el mundo que ha descargado las librerías contribuidas por usuarios Arduino, Me gustaría agradecer a los autores quienes generosamente han compartido su conocimiento. La disponibilidad de un amplio alcance de hardware es un grande parte de lo que hace a Arduino emocionante, gracias a los proveedores por disponibilidad y soporte a un vasto alcance de grandes aparatos. Los siguientes fueron útil en proporcionar hardware usado en el libro: SparkFun, Creador Shed, Gravitech, y NKC Electronics. Otros proveedores que han sido útiles incluyen Moderno Dispositivo, Liquidware, Adafruit, MakerBot Industries, Mindkits, Oomlout, y SK Pang. Nick le gustaría agradecer a todo el mundo que estuvo involucrado con Tinker London, sobre todo a Alexandra, Peter, Brock Craft, Daniel Soltis y todas las personas quienes asistieron en talleres en los últimos años. Nick finalmente agradece a su familia, Jeanie, Emily, y Finn, quien estuvo de acuerdo en dejarle hacer esto en sus vacaciones de verano, y por supuesto, mucho después de lo que ellos habían pensado, y a su padres, Franco y Eva, por llevarlo hasta a tomar cosas aparte. Último pero no menos, Yo expresar gracias a las siguientes personas: Joshua Noble por introducirme a O'Reilly. Su libro, Programming Interactivity, es altamente recomendado para aquellos interesados en ampliar su conocimiento en la computación interactiva. Robert Lacy-Thompson por afrecer asesoramiento temprano con la primera edición. Mark Margolis por su apoyo y ayuda como una caja de resonancia en la concepción y desarrollo del libro. Prólogo xix Agradezco a mis padres por ayudarme a ver que las artes creativas y tecnología no fueron distintas entidades y que, cuando combinada, ellas pueden conducir a extraordinarios resultados. Y finalmente, este libro no habría sido comenzado o terminado sin el apoyo de mi esposa, Bárbara Faden. Mi agradecido reconocimiento para ella por mantenerme motivado y por su lectura cuidadosa y contribuciones para el manuscrito. Notas en la Segunda Edición La segunda edición de este libro ha seguido relativamente rápidamente desde la primera, impulsado por la version de Arduino 1.0. El propósito declarado de 1.0 es introducir un cambio significativo va a allanar el camino para futuras mejoras pero cambiando algunos códigos escritos para software anterior. Estos han requerido cambios en código en muchos de los capítulos de este libro. Más cambiado es el Capítulo 15, Ethernet y Networking, y Capítulo 13, Comunicación Usando I2C y SPI, pero todo de las recetas en esta edición ha sido migrado a 1.0, con muchas actualizaciones para utilizar características nuevas en esta versión. Si usted está utilizando una versión anterior a Arduino 1.0, entonces usted puede descargar código de la primera edición de este libro. Ver " Plataforma Arduino Notas de versión" en página xv para descargar detalles. Apéndice H, Migración a Arduino 1. 0, ha sido adicionado para describir los cambios introducidos por Arduino versión 1.0. Este describe cómo actualizar código antiguo para utilizar con Arduino 1.0. Recetas para dispositivos que no son más extensamente disponibles han sido actualizadas para utilizar reemplazos actuales y algunos nuevos sensores y dispositivos inalámbricos han sido añadidos. Erratas publicadas en el sitio de O'Reilly ha sido corregida, gracias a lectores que se han tomado tiempo para notificarnos de éstas. Creemos que le gustarán las mejoras hechas en Arduino 1.0, así como las mejoras hechas para esta edición de Arduino Cookbook. La primera edición fue bien recibida; la crítica constructiva es dividida entre las personas que prefirieron más contenido técnico y aquellos que prefirieron menos. En un libro que nosotros limitado a sólo 700 páginas (para mantener lo asequible y portátil), que parece indicar que el equilibrio adecuado ha sido alcanzado. Prólogo xx CAPÍTULO 1 Empezando 1.0 Introducción El entorno de Arduino ha sido diseñado para ser de fácil uso para principiantes que no tienen experiencia en software o electrónica. Con Arduino, usted puede construir objetos que pueden responder a y/o controlar luz, sonido, tocar, y movimiento. Arduino ha sido usado para crear una asombrosa variedad de cosas, incluyendo instrumentos musicales, los robots, esculturas de luz, juegos, muebles interactivos, e incluso prendas de vestir interactivas. Si usted no es un principiante, por favor siéntase libre de adelantarse a recetas que le interesen. Arduino es usado en muchos programas educativos en todo el mundo, particularmente por diseñadores y artistas que quieren fácilmente crear prototipos pero hacer sin necesitar una profunda comprensión de los detalles técnicos detrás sus creaciones. Porque este es diseñado para ser utilizado por gente no técnica, el software incluye muchos ejemplos de código para demostrar cómo utilizar las varias facilidades de la tarjeta Arduino. Aunque este es fácil a utilizar, el hardware subyacente de Arduino trabaja al mismo nivel de sofisticación que los ingenieros emplear para construir dispositivos embebidos. Personas que ya han con microcontroladores son también han sido atraído por Arduino por su ágil capacidad de desarrollo y su facilidad para rápida implementación de ideas. Arduino es mejor conocido por su de hardware, pero usted también necesita software para programar dicho hardware. Ambos el hardware y la software se llaman "Arduino". La combinación le permite a usted crear proyectos que sienten y controlan el mundo físico. El software es gratuito, de código abierto, y multiplataforma. Las tarjetas son baratas, o usted puede construir su propia tarjeta (El diseño de hardware también es de código abierto). Además, hay es una activa comunidad Arduino como apoyo, esta accesible mundialmente a través de los foros y wikis de Arduino (Conocido como la Zona de juegos Arduino). Los foros y las 1 wikis ofrecen ejemplos de proyectos en desarrollo y soluciones par problemas que pueden servir de inspiración y asistencia sobre como usted realizar sus propios proyectos. Las recetas en este capítulo comenzarán por explicar cómo instalar el entorno de desarrollo y cómo a compilar y ejecutar un ejemplo de Sketch. El código fuente en el que contiene las instrucciones para controlar Arduino funcionalmente es generalmente referido como un Sketch en la comunidad Arduino. La palabra Sketch será usado a lo largo de este libro para remitir a un Programa de código Arduino. El Skecth Blink, que viene con Arduino, es usado como un ejemplo para recetas en este capítulo, aunque la última receta en el capítulo va además por añadir sonido y reúne entradas a través de algunos hardware adicionales, no sólo el parpadeo de la luz integrada en la tarjeta. El Capítulo 2 incluye cómo estructurar un Sketch para Arduino y proporciona una introducción a la programación. Si usted ya sabe las formas básicas acerca de Arduino, siéntase libre de saltarse hacia próximos capítulos. Si es su primera vez usando Arduino, tenga paciencia en estas primeras recetas, tendrá más suaves resultados más tarde. Arduino Software Programas, llamados Sketches, son creados en un ordenador usando el entorno integrado de desarrollo (IDE) Arduino. El IDE permite usted a escribir, editar código y convertir este código en instrucciones que el hardware Arduino entiende. El IDE también transfiere aquellas instrucciones a la tarjeta Arduino (Un proceso llamado carga). Arduino Hardware La tarjeta Arduino es donde el código que usted escribe es ejecutado. La tarjeta puede sólo controlar y responder a la electricidad, así componentes específicos son adjuntados para permitir interactuar con el mundo real. Estos componentes pueden ser sensores, que convierten algún aspecto del mundo físico a electricidad así que la tarjeta puede sientir esto, o actuadores, que consiguen electricidad desde la tarjeta y la convierten que cambia el mundo físico. Los ejemplos de sensores incluyen interruptores, acelerómetros, y sensores ultrasónicos de distancia. Los actuadores son cosas como luces y LEDs, altavoces, motores, y pantallas. Hay una variedad de tarjetas oficiales que usted puede utilizar con el software Arduino y una amplia gama de tarjetas compatibles con Arduino producidas por miembros de la comunidad. Las tarjetas más populares tienen conector USB que se utiliza para proporcionar potencia y conectividad para cargar su software en la tarjeta. La Figura 1-1 muestra una tarjeta básica con la que inicia la mayoría de personas, la Arduino Uno. 2 Figura 1-1. Tarjeta Básica: Arduino Uno. Fotografía cortesía de todo.to.it. Arduino Uno tiene un segundo microcontrolador a bordo para manejar toda la comunicación USB; el pequeño chip montado en superficie (El ATmega8U2) es situado cerca del zócalo USB en la tarjeta. Este puede ser programado por separado para permitir a la tablero para aparecer como diferentes dispositivos USB (Véase Receta 18.14 para un ejemplo). La tarjeta Arduino Leonardo reemplaza los controladores ATmega8U2 y la ATmega328 con un solo chip ATMEGA32U4 que implementa el protocolo USB en software. Las tarjetas compatibles con Arduino Teensy y Teensy de PJRC (también capaces de emular dispositivos USB. Tarjetas más antiguas, y más de las tarjetas compatibles con Arduino, utilizan un chip de la empresa FTDI que proporciona una solución de hardware USB para conexión al puerto serial de su ordenador. Usted puede conseguir tarjetas tan pequeñas como una estampilla, tal como la Arduino Mini y Pro Mini; las tarjetas más grandes tienen más opciones de conexión más potentes procesadores, tal como la Arduino Mega; y tarjetas adaptadas para aplicaciones específicas, tal como la LilyPad para aplicaciones utilizables, la Fio proyectos inalámbricos, y la Arduino Pro para aplicaciones embebidas (Proyectos independientes que a menudo funcionan con baterías). Reciente adiciones al alcance incluiyen la Arduino ADK, que tiene un zócalo USB anfitrión sobre este y que es compatible con la Android Open Accessory Development Kit, el oficialmente aprobado método de adjunción de hardware para dispositivos Android. La tarjeta Leonardo usa un chip controlador (El ATMEGA32U4) que es capaz de presentarse a sí mismo como varios dispositivos HID. 3 La tarjeta Ethernet incluye conectividad Ethernet, y tiene una opción Power Over Ethernet, así es posible utilizar un solo cable para conectar y energizar la tarjeta. Otras tarjetas compatibles con Arduino también están disponibles, incluyendo las siguientes: Arduino Nano, una tarjeta diminuta con capacidad USB, de Gravitech (itech.us/arna30wiatn.html) Bare Bones Board, una tarjeta de bajo costo disponible con o sin capacidad USB, de Modern Device (Boarduino, un tarjeta protoboard compatible de bajo costo b, de Adafruit Industrias (Seeeduino, un flexible variación de la tarjeta USB estándar, de Seeed Estudio Bazar (Teensy y Teensy + +, tarjetas diminutas pero extremadamente versátiles, de PJRC (/ teensy /) Una lista de tarjetas . compatibles con Arduino está disponible en Vea también Una visión de conjunto . de tarjetas Arduino: Guías online para comenzar con Arduino están disponible en de/Windows para Windows, para Mac OS X, y /Learning/Linux para Linux. Una lista de más de cien tarjetas que pueden ser usadas con el Entorno de Desarrollo Arduino puede encontrarse en: inocompatible.html 1.1 Instalando el Entorno de Desarrollo Integrado (IDE) Problema Usted quiere instalar el Entorno de Desarrollo Arduino desarrollo en su ordenador. Solución El software Arduino para Windows, Mac, y Linux puede ser descargado desde http:// arduino.cc /en/Main/Software. La descarga de Windows es un archivo ZIP. Descomprimir el archivo para cualquier conveniente el directorio Archivos de programa/Arduino es un lugar sensato. 4 El software gratuito para descomprimir archivos, llamado 7-Zip, puede ser descargado desde . Descomprimir el archivo creara una carpeta llamada Arduino-00 (Donde es el número de versión de Arduino que usted descargó). El directorio contiene el archivo ejecutable (Llamado Arduino.exe), a lado de varios otros archivos y carpetas. Haga doble clic en el archivo Arduino.exe y la pantalla de inicio debería aparecer (Véase Figura 1-2), seguido por la ventana principal del programa (Véase

comunes construcciones del lenguaje, y proporciona una visión general de la sintaxis para lectores que no están familiarizados con C o C + +, el lenguaje que utiliza Arduino. Ya que hacer ejemplos interesantes requiere hacer que Arduino realice algo, las recetas utilizan capacidades físicas de la tarjeta que son detallados en próximos capítulos. Si cualquier código en este capítulo no está claro, siéntase libre adelantarse, particularmente al Capítulo 4 para más sobre salida serial y el Capítulo 5 para más sobre uso de pines digitales y analógicos. Usted no necesita entender todo el código en los ejemplos, sin embargo, ve cómo funcionan capacidades específicas que son enfocadas en las recetas. Aquí hay algunas de las funciones más comunes, usadas en los ejemplos que son incluidos en los próximos capítulos: Serial.println(valor); Imprime el valor al Monitor Serial del IDE Arduino así usted podrá ver la Salida de Arduino en su computador; ver la Receta 4.1. pinMode(pin, mode); Configura un pin digital para leer (INPUT) o escribir (OUTPUT) un valor digital; ver la introducción en el Capítulo 5. digitalRead(pin); Lee un valor digital (HIGH o LOW) en un pin para establecerlo como entrada; ver la Receta 5.1. digitalWrite(pin, valor); Escribe un valor digital (HIGH o LOW) en un pin para establecerlo como salida; ver la Receta 5.1. 22 2.1 Estructurando un Programa Arduino Problema Usted es nuevo en programación y quiere entender la construcción de un Programa de Arduino. Solución Programas para Arduino son en generalmente llamados Sketches; los primeros usuarios eran artistas y diseñadores y los Sketches reflejan la forma rápida y fácil para realizar una idea. Los términos Sketches y programa son intercambiables. Los Sketches contienen el código que tiene las instrucciones que la tarjeta llevará a cabo. El Código que necesita ejecutar sólo una vez (como configurar la placa para su aplicación) necesita ser colocado en la función setup. El código a ser ejecutar continuamente después de la configuración irá en la función loop. Aquí un típico Sketch: const int ledPin = 13; // LED conectado al pin digital 13 // El método setup () se ejecuta una vez, cuando el Sketch se inicia void setup () { pinMode (ledPin, OUTPUT); // inicializa el pin digital como salida } // El método loop () se ejecuta una y otra vez, void loop () { digitalWrite (ledPin, HIGH); // encender el LED delay (1000); // esperar un segundo digitalWrite (ledPin, LOW); // apagar el LED delay (1000); // esperar un segundo } Cuando el IDE de Arduino IDE termina de cargar el código, y cada vez que usted potencie la tarjeta después de haber subido este código, este iniciará y llevara a cabo las instrucciones secuencialmente. Ete ejecuta el código una vez en setup y entonces va a través del código en loop. Cuando llega al final del loop (Marcada por la llave cerrada, }) este regresa al comienzo del loop. Discusión Este ejemplo continuamente parpadea un LED por escritura de salidas HIGH y LOW en un pin. Ver el Capítulo 5 para aprender más acerca del uso de los pines de Arduino. Cuando el Sketch comienza, el código en setup configura el modo del pin (lo que permite iluminar un LED). Después que el código en setup es completado, el código en loop es repetidamente llamado (para parpadear el LED) siempre y cuando la tarjeta Arduino este potenciada. 23 Usted no necesita saber esto para escribir Sketches Arduino, pero programadores con experiencia en C y C+ + pueden preguntarse a donde se ha ido la esperada función main(). Está ahí, pero está oculta bajo la cubierta del Entorno de Arduino. El proceso de construcción de crea un archivo intermedio que incluye el código del Sketch y las siguientes declaraciones adicionales: int main (void) { init (); setup (); for (; ;) loop (); return 0; } La primera cosa que sucede es una llamada a una función init() que inicializa el Hardware Arduino. Siguiente, la función setup() de su Sketch es llamada. Finalmente, la función loop() es llamado una y una vez. Porque el ciclo for nunca termina, la sentencia return es ejecutada. Ver También La Receta 1.4 explica cómo Cargar un Sketch a la tarjeta Arduino. El Capítulo 17 y más información sobre el proceso de construcción. 2.2 Uso de Tipos Primitivos Simples (Variables) Problema Arduino tiene diferente tipos de variables para representar valores eficientemente. Usted quiere saber cómo seleccionar y utilizar estos tipos de datos de Arduino. Solución Aunque el tipo de dato int (Abreviatura de número entero, un valor de 16 bits en Arduino) es la más común elección para los valores numéricos que se encuentran en las aplicaciones de Arduino, usted puede utilizar la Tabla 2-1 para determinar los tipos de datos que espera que encajen en su aplicación. 24 Tabla 2-1. Tipos de datos Arduino Tipos Numéricos Bytes Rango Uso int unsigned int int. long Unsigned long float double float boolean falso double float verdadero char byte char, Otros tipos Uso String void Representa cadenas de chars (Caracteres) normalmente se utiliza para contener texto Sólo se utiliza en las declaraciones de funciones donde no se devuelve ningún valor. Discusión Excepto en situaciones donde máximo rendimiento o eficiente memoria son requeridos, variables declaradas usando int serán adecuadas para valores numérico si los valores hacer no exceden su alcance (Mostrado en la primera fila en Tabla 2-1) y si usted no necesita a trabajar con valores fraccionarios. Muchos de los ejemplos oficiales de Arduino declaran variables numéricas como int. Pero a veces usted necesitara elegir un tipo que específicamente se ajuste a su aplicación. A veces usted necesita números negativos y a veces no, por lo que los tipos numéricos vienen en dos versiones: signed y unsigned. Valores unsigned son siempre positivos. Variables sin la palabra clave unsigned son signed así que ellos pueden representar valores negativos y positivos. Uno razón para utilizar valores unsigned es cuando la gama de valores signed no encajarán en el rango de la variable (Una variable unsigned tiene dos veces la capacidad que una variable signed). Otra razón por la que los programadores eligen utilizar tipos unsigned es a claramente indicar a las personas que el código esperado nunca será un número negativo. Los tipos boolean tienen dos posibles valores: true o false. Estos son comúnmente usados en revisión del estado de un interruptor (Si es pulsado o no). Usted puede también utilizar HIGH y LOW como equivalentes a true y false donde este tiene más sentido; digitalWritte(pin, HIGH)forma es una más expresiva de encender un LED que digitalWrite(pin, true) o digitalWrite(pin, 1), aunque todo de estos son tratados idénticamente cuando el Sketch es ejecutado, dibujo en realidad carreras, y probablemente vienen todas estas formas de código publicados en la web. 25 Vea también La referencia de Arduino en proporciona detalles sobre tipos de datos. 2.3 Usando Números de Punto Flotante Problema Los números de punto flotante son usados para valores expresados con punto decimal (Esta es la forma para representar valores fraccionarios). Usted quiere calcular y comparar estos valores en su Sketch. Solución El siguiente código muestra cómo a declarar variables de punto flotante, ilustrando problemas que puedan encontrarse comparaciones de valores de punto flotante, y demuestra cómo superarlos: /* * Ejemplo de punto flotante * Este Sketch inicializa un valor flotante a 1,1 * Se reduce varias veces el valor de 0,1 hasta que el valor sea 0 * / float valor = 1,1; void setup () { Serial.begin (9600); } void loop () { valor = valor - 0.1 / / reducir el valor de 0,1 en cada iteración del loop if (valor == 0) Serial.println ("El valor es exactamente cero"); else if (almostEqual(valor, 0)) { Serial.print ("El valor"); Serial.print (valor, 7); // para imprimir 7 lugares decimales Serial.println ("es casi igual a cero"); } else Serial.println(valor); delay (100); 26 } // Devuelve un valor true si la diferencia entre a // Cambia el valor de DELTA a la diferencia máxima igualad booleana almostEqual(float a, float b) { float const DELTA = .0001 / / m{axima diferencia igual if (a == 0) return fabs(b) > = 2; valor = valor >> 2; // desplaza el valor dos lugares a la derecha