



Universidad Nacional de La Matanza

DEPARTAMENTO DE INGENIERÍA E INVESTIGACIONES TECNOLÓGICAS

SISTEMAS OPERATIVOS AVANZADOS

Proyecto: Plataforma Autonivelable

Sistemas Embebidos (IoT) y Android

Ciclo Lectivo

2° Cuatrimestre 2018

Comisión

Miércoles - Noche

Docentes

Lic. De Luca, Graciela

Ing. Carnuccio, Esteban

Ing. García, Gerardo

Ing. Valiente, Waldo

Ing. Volker, Mariano

Alumnos

Abdala, Nahuel DNI: 37181169

Cáceres, Silvina DNI: 23834949

Dórtona, Agustín DNI: 36714539

Pintos, Leandro DNI: 35901158

Porro, Diego DNI: 37216260

Contenido

OBJETIVO DEL PROYECTO	3
SOLUCION PROPUESTA PARA EL PROYECTO	3
MODO DE FUNCIONAMIENTO	3
HARDWARE UTILIZADO	3
SOFTWARE UTILIZADO.....	3
DESCRIPCION.....	4
DIAGRAMAS	6
PROBLEMAS QUE SURGIERON DURANTE EL DESARROLLO DEL PROYECTO	8
APLICACIÓN ANDROID (PLATAFORMA AUTONIVELABLE).....	9
DIAGRAMA DE FUNCIONES (SOFTWARE)	11

OBJETIVO DEL PROYECTO

Nuestro objetivo es desarrollar un sistema embebido (SE) creando una plataforma que se autonivele dependiendo la superficie donde se encuentra apoyada.

SOLUCION PROPUESTA PARA EL PROYECTO

Nos enfocamos en utilizar un sistema embebido como Arduino Mega con sensores apropiados y la programación necesaria para que realice la funcionalidad esperada. Además, este sistema embebido se comunicará a través de un módulo Bluetooth con un dispositivo Android para que el usuario pueda obtener información del estado de la plataforma autonivelable.

MODO DE FUNCIONAMIENTO

El sistema presenta un nivel de inclinación inicial de la plataforma respecto a la superficie donde se encuentre apoyada, a partir del cual reaccionará ajustando sus patas hasta que la superficie de la plataforma logre alcanzar el nivel deseado.

HARDWARE UTILIZADO

- Materiales
 - o 1 placa Arduino Mega
 - o Resistencias 2k ohm
 - o 2 puente H L293D
 - o 3 patas, compuestas cada una por: un cilindro, un pistón, un sinfín y un vástago.
 - o 1 módulo Bluetooth para la placa Arduino modelo HC -05(maestro-esclavo)
 - o Cables varios HH y MM
 - o 1 Protoboard de 830 puntos
 - o 1 Protoboard de 400 puntos
 - o 3 transformador de 2A y 5V.
 - o 1 superficie de madera fibrofacil
 - o Estaño
- Sensores
 - o 1 Sensor Acelerómetro MPU 6050
 - o 1 Sensor de distancia Ultrasonido HC-SR04
 - o 3 pulsadores
 - o 3 contactos (simil-pulsadores)
- Actuadores
 - o 3 motores DC 6V
 - o 3 diodo Led
- Sensores Android
 - o Acelerómetro
 - o Sensor de Proximidad
 - o Sensor de luminosidad

SOFTWARE UTILIZADO

- IDE Arduino
- Android Studio
- Versión de Android 5.0

DESCRIPCION

Detalles técnicos de los sensores utilizados

Sensor acelerómetro MPU 6050

El MPU6050 es un sensor de movimiento que posee un giroscopio en sus tres ejes (x,y,z) con una alta precisión. internos de 16 bit y se maneja por I2C desde cualquier microcontrolador como Arduino. Dentro de una de las ventajas mencionar, el MPU 6050 posee conversores analógicos cada uno de los ejes de cada uno de los sensores para obtener simultáneo con un rango de hasta 2000°/s para el giroscopio y para el acelerómetro.



acelerómetro y
Posee ADC

que cabe
digitales por
los valores en
hasta +-16g

Uso en el sistema embebido

Permite determinar si la plataforma está nivelada a través del análisis de los valores de coordenadas X e Y ignorando los valores arrojados por el sensor para el eje Z.

Pulsador

Botón pulsador 6*6*5 mm 4-pin Dip de encendido/apagado

Uso en el sistema embebido

Detectar si la plataforma está apoyada.



Sensor de distancia Ultrasonido Sr04

Los sensores de ultrasonido son muy útiles para medir distancias y detectar obstáculos.

El funcionamiento es simple, envía una señal ultrasónica inaudible y nos entrega el tiempo que demoró en ir y venir hasta el obstáculo más cercano que detectó.

Generalmente, están conformados por dos cilindros puestos uno al lado del otro, uno de ellos es quien emite la señal ultrasónica, mientras que el otro es quien la recibe.

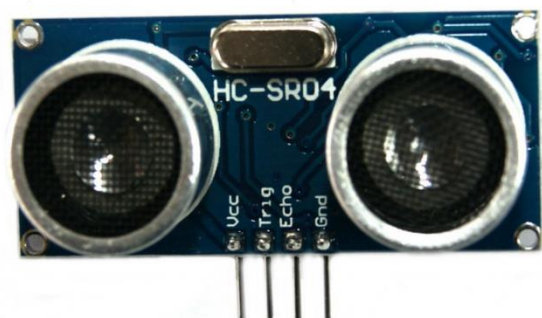
El sensor Sr04 en particular tiene una sensibilidad muy buena del orden de los 3 mm, podemos decir que su sensibilidad es muy buena.

El sensor Sr04 cuenta el tiempo que transcurre entre la emisión y la recepción de la señal ultrasónica, podemos deducir que el tiempo es dependiente de la distancia, la señal tardará más en ir y volver si el objeto está lejos que si está cerca.

El sensor Sr04 tiene 4 pines, uno es Vcc, otro es GND y un pin de trigger donde enviamos un pulso al sensor para inicializarlo, y otro pin más Echo donde nos viene el resultado final a la placa microcontroladora Arduino.

Uso en el sistema embebido

Medir la altura de la plataforma para permitir ajustarla desde la aplicación mobile.



Detalles técnicos de los actuadores utilizados

Motor DC 6V

Características técnicas:

Voltaje de operación: 3V - 6V

Velocidad Angular Nominal: 125 RPM

Reducción: 1:48

Consumo Máximo de corriente: 150 mA

Peso: 50g

Ruido: < 65Db

Este motor DC posee una caja reductora integrada que le permite entregar un buen torque en un pequeño tamaño y bajo voltaje. Su caja posee varios engranajes en su interior que posibilita la reducción y potencia así como estabilidad de la velocidad.

Función operativa en el sistema embebido

Ajustar la altura de las patas para afectar el nivel de la plataforma.



Diodo led

Un diodo led es un diodo que además de de la corriente solo en un sentido, en el sentido corriente pasa por el diodo, este emite luz.

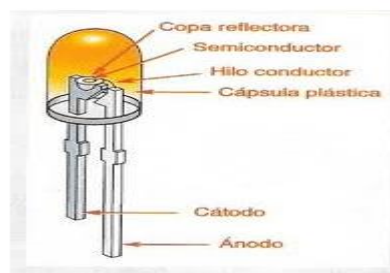
Características técnicas:

Los diodos leds tienen dos patas de conexión, corta. Para que pase la corriente y emita luz se pata larga al polo positivo y la corta al negativo.

Función en el sistema embebido

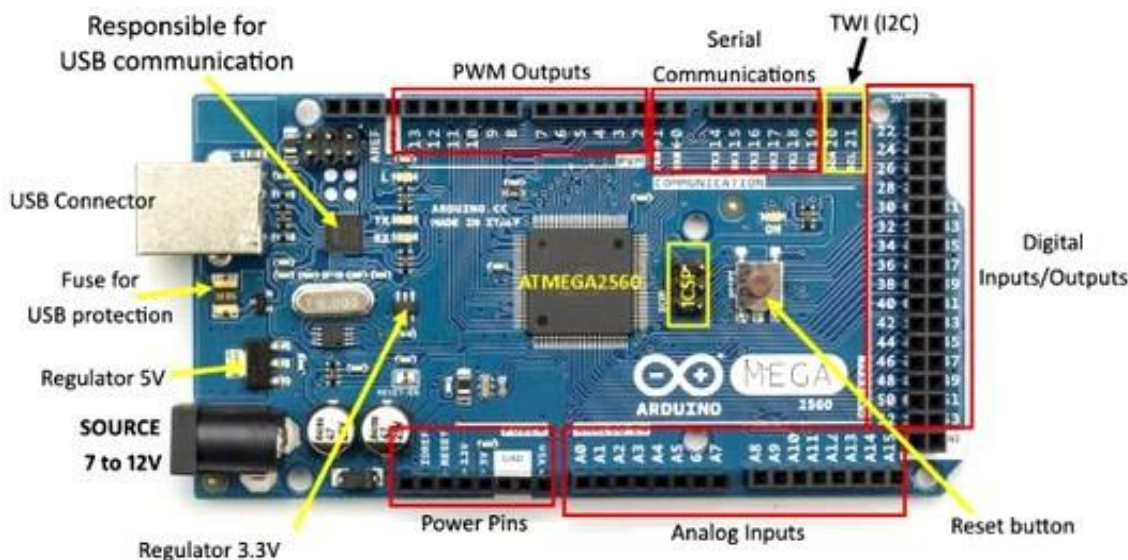
Emitir luz verde cuando la plataforma esté cuando está nivelando.

Prende una luz azulada sobre la superficie cuando la luminosidad detectada en el dispositivo Android es baja.



Arduino MEGA

Características



Arduino Mega es el microcontrolador de la familia Arduino, posee 54 pines digitales que funcionan como entrada/salida, 16 entradas analógicas, un cristal oscilador de 16 Mhz, una conexión USB, un botón de reset y una entrada para la alimentación de la placa.

La comunicación entre la computadora y Arduino se produce a través del puerto Serie. Posee un convertidor usb-serie, por lo que solo se necesita conectar el dispositivo a la computadora utilizando un cable USB.

Arduino Mega posee las siguientes especificaciones:

- Microcontrolador: ATmega 2560
- Voltaje Operativo: 5V
- Voltaje de Entrada: 7V-12V
- Voltaje de Entrada(Limites): 6V-20V
- Pines Digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
- Pines análogos de entrada: 16
- Corriente DC por cada pin Entrada/Salida: 40mA
- Corriente DC Entregada en el pin 3.3V: 50mA
- Memoria Flash: 256 KB (8 KB usados para el bootloader)
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 Mhz

BLUETOOTH HC05

El módulo Bluetooth HC05 permite ser configurado para operar como maestro o esclavo.

- Modo maestro: genera conexiones hacia otros dispositivos Bluetooth.
- Modo esclavo: recibe peticiones de conexión.

Características:

Compatible con el protocolo Bluetooth V2.0

Voltaje de Alimentación: 3.3V – 6V

Interfaz: UART RS 232 Serial

Alcance: 10 Mts

Tasa de Velocidad de Modulación en Baudios por defecto: 9600 bps

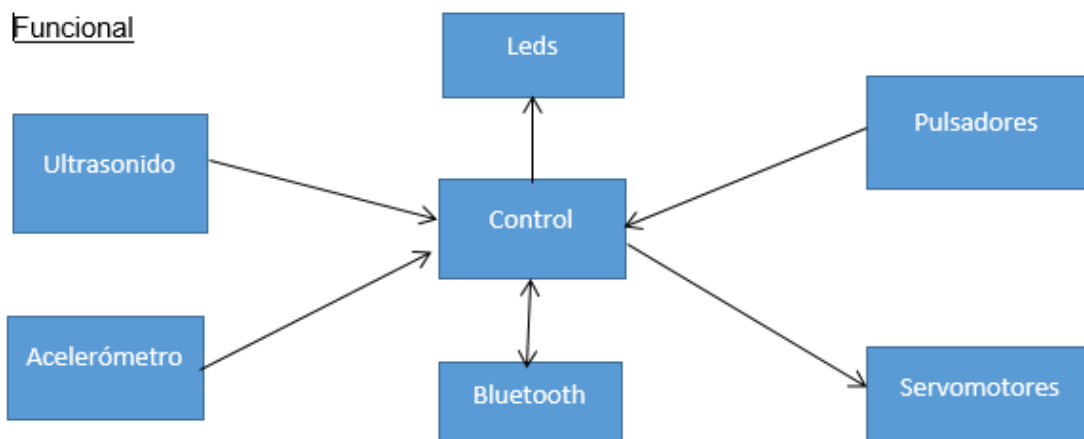
Función en el sistema embebido

Responsable de establecer la comunicación con el dispositivo Android que a través de la aplicación mobile desarrolla permitirá, establecer el modo “nivelación” o manual con el cual se podrá establecer la altura de la plataforma (acelerómetro del celular), encender la lámpara (luminosidad del celular) y subir o bajar la plataforma por completo (proximidad del celular).

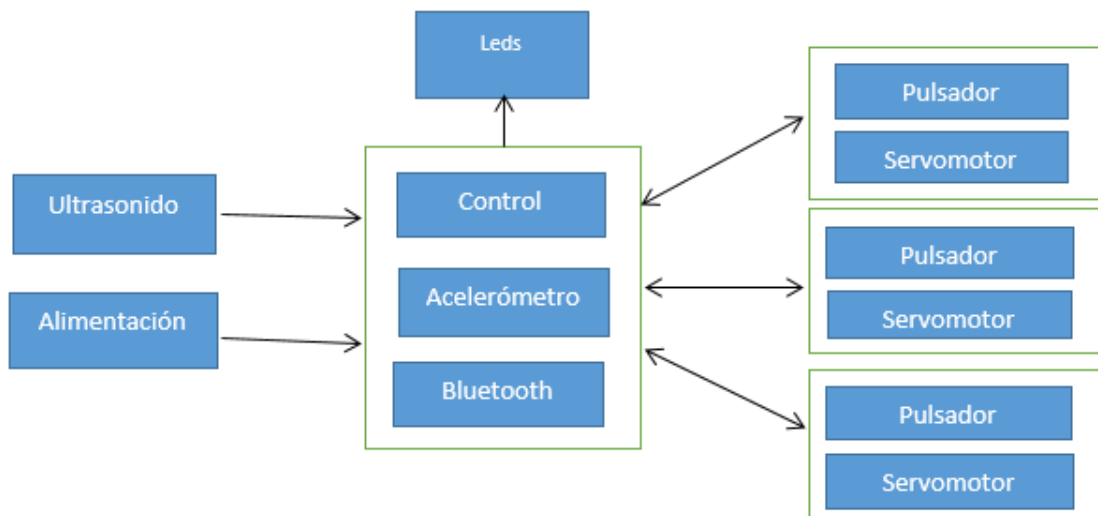


DIAGRAMAS

Funcional



Físico



Estados

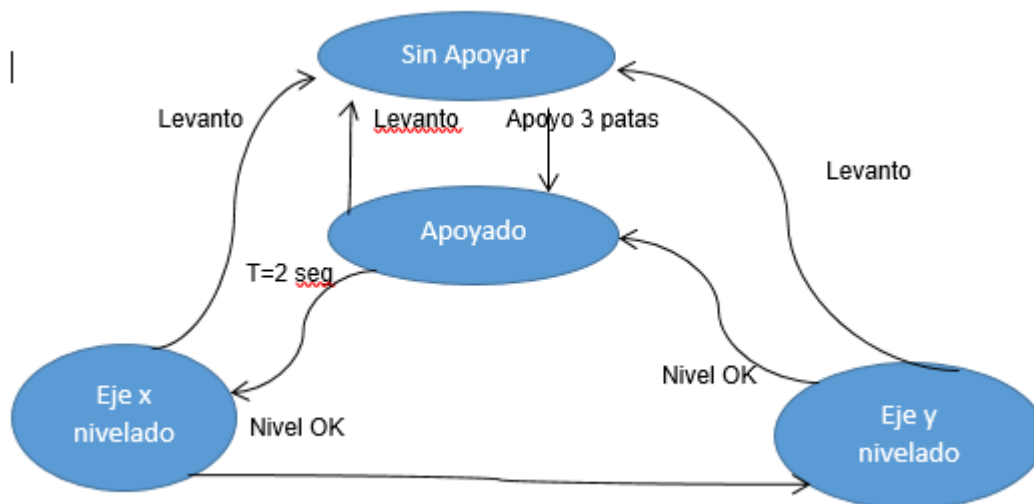
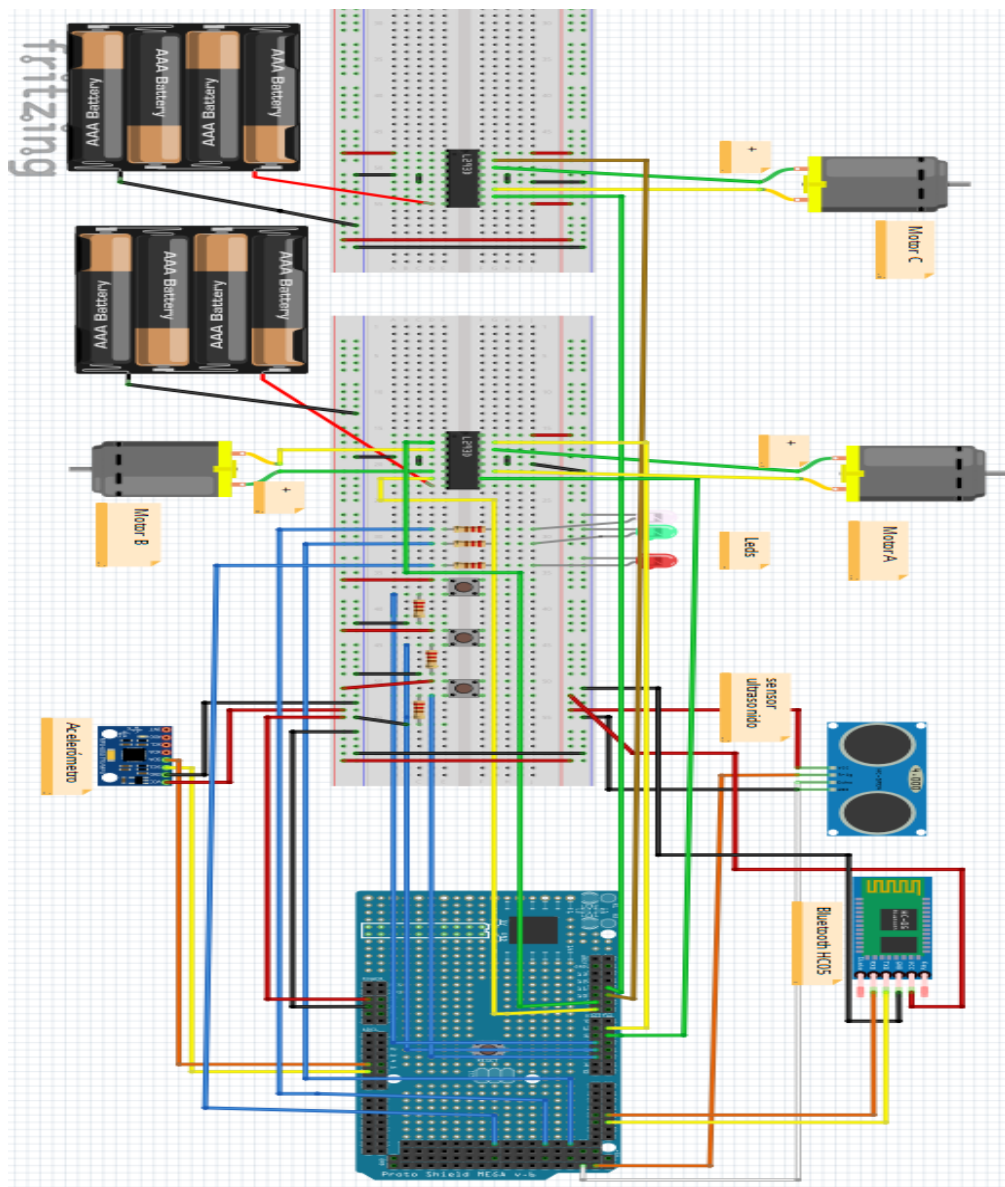


Diagrama de Conexión



PROBLEMAS QUE SURGIERON DURANTE EL DESARROLLO DEL PROYECTO

Circuito

Teniendo los tres motores conectados directamente a la salida del Arduino MEGA (5V 20mA) nos encontramos con que no se podían girar los tres motores a la vez. Se solucionó conectando cada motor a un transformador independiente (5V 2A).

Algoritmo

En principio el orden de nivelación era primero nivelar el eje X y posteriormente nivelar el eje Y. Al hacer esto vimos que por la distribución de las patas de la plataforma al intentar nivelar el eje Y, provocábamos que se desnivele nuevamente el eje X considerablemente. Esto se solucionó invirtiendo

el orden de nivelación pasando primero a nivelar el eje Y , y posteriormente nivelar el eje X en cuyo último proceso se chequea constantemente que no se haya desnivelado el eje Y ya que en tal caso se debe volver a nivelar éste.

Android

Inicialmente se pensaba implementar el pasaje de mensajes utilizando formato JSON, pero desistimos de ese formato debido a la complejidad que implicaba en la implementación en comparación con los mensajes que se necesitaban enviar. Por lo tanto, se decidió enviar mensajes por texto plano.

APLICACIÓN ANDROID (PLATAFORMA AUTONIVELABLE)

Una vez hechas las conexiones en el sistema embebido, pasamos a realizar la implementación de la aplicación en Android.

Sensores utilizados

Sensor	Aplicación
Acelerómetro	Por medio de la inclinación que se aplique al dispositivo, se ordenara a la plataforma la altura a la que colocarse.
Proximidad	Funciona como una variable booleana en donde la primera vez que se detecta el evento, la plataforma se coloca a la altura mínima, al volverse a detectar, la plataforma se coloca a la altura máxima. Este comportamiento se repite cada vez que se detecte el evento.
Luminosidad	Al detectar falta de luz en el contexto del dispositivo Android (se entiende que será el mismo que el de la plataforma), el mismo le ordenará a la plataforma que encienda la lámpara.

Interfaz de usuario



Activities

- **MainActivity:** se realizan los métodos necesarios para efectuar la conexión con Bluetooth y verifica su estado (“encendido” o “apagado”).
- **Activity_comunicacion:** muestra el estado de los sensores, la altura actual de la plataforma, crea el Bluetooth socket para comunicar con el sistema embebido y crea un hilo secundario que recibe y envía datos por medio del Bluetooth (comunicación con embebido), los cuales serán procesados ya sea por parte de la aplicación Android (datos recibidos) o por parte del embebido (datos enviados).

Broadcast Receiver

- **PairingRequest:** se utiliza en caso de que el dispositivo Android no se encuentre emparejado con el embebido para proveer la clave de emparejamiento (123456) en caso de que el usuario la desconozca.

DIAGRAMA DE FUNCIONES (SOFTWARE)

