



Universidad Nacional de La Matanza

Departamento de Ingeniería e Investigaciones Tecnológicas

Sistemas Operativos Avanzados

Sistemas Embebidos

Internet de las cosas

Android

BALANZA DIGITAL CON VOLCADO AUTOMÁTICO

Integrantes:	Ferreira, Jonathan Iván	33960677
	Lavarello, Matías	37560619
	Merino, Matías	35970494
	Montañés, Alfredo	35131585

Primer cuatrimestre 2017

1. OBJETIVOS DEL PROYECTO	2
2. ENTORNO DE DESARROLLO	3
HARDWARE	3
<i>Sistema embebido</i>	3
<i>Aplicación Android</i>	3
SOFTWARE.....	3
<i>Sistema embebido</i>	3
<i>Aplicación Android</i>	3
3. DIAGRAMA DE CONEXIÓN	4
4. IMPLEMENTACIÓN	5
SISTEMA EMBEBIDO.....	5
<i>Celda de carga</i>	5
<i>Motor DC</i>	5
<i>Servo Motor</i>	5
<i>Módulo Bluetooth HC05</i>	5
APLICACIÓN ANDROID.....	6
<i>Actividades</i>	6
<i>Sensores</i>	6
Acelerómetro.....	6
Proximidad.....	6
Orientación	6
<i>Bluetooth</i>	6
5. FUNCIONAMIENTO.....	7
SISTEMA EMBEBIDO.....	7
APLICACIÓN ANDROID.....	7
6. PRODUCTO TERMINADO	8
SISTEMA EMBEBIDO.....	8
APLICACIÓN ANDROID.....	8
7. BIBLIOGRAFÍA.....	9



1. Objetivos del proyecto

Aplicar los conocimientos sobre Internet de las cosas, Sistemas Embebidos y Android aprendidos durante la cursada de la asignatura Sistemas Operativos Avanzados, desarrollando una balanza digital de precisión que permita detallar cuantos elementos tiene sobre ella y los vuelque para un embolsado automático al llegar a determinada cantidad.

2. Entorno de desarrollo

Hardware

Sistema embebido

- Notebook Intel i5, 8GB, Windows 7.
- 1 x placa Arduino Uno
- 1 x módulo Bluetooth HC05
- 1 x motor caja reductora de 3 a 6 Volts Cc
- 1 x servo motor
- 1 x celda de carga de 1Kg.
- 1 x transmisor de celda de carga HX711
- 1 x transistor
- 1 x protoboard
- Cables de conexión

Aplicación Android

- Notebook Intel i5, 8GB, Windows 7
- Smartphone Moto G 5 Plus
- Sensores: acelerómetro, orientación, proximidad.

Software

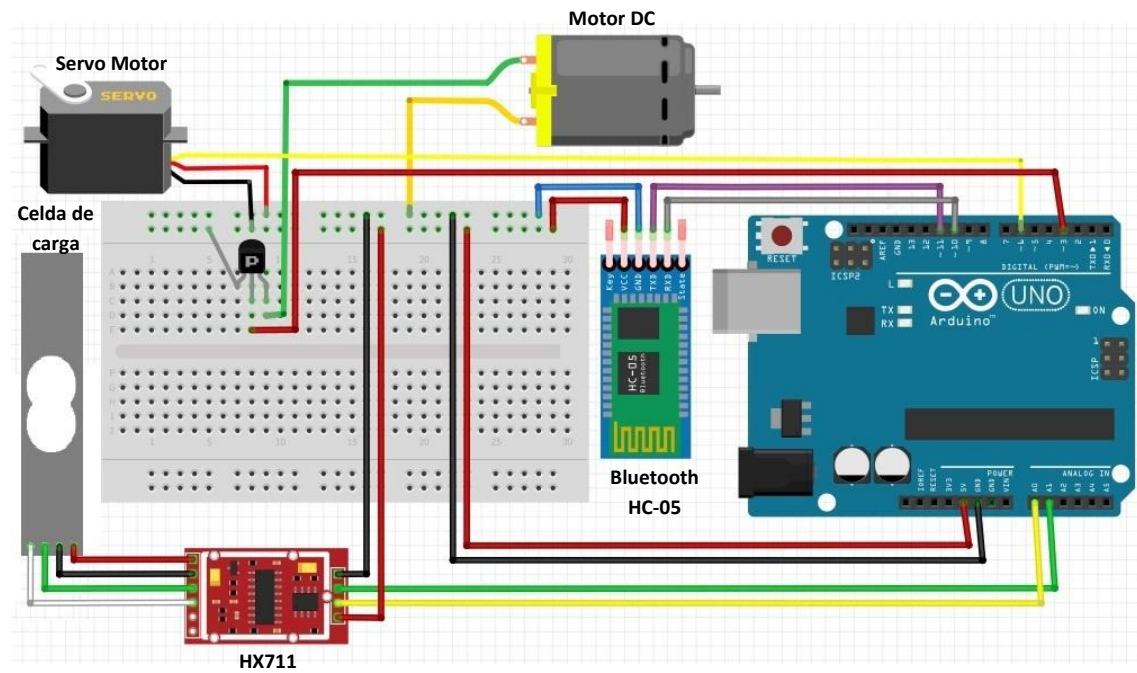
Sistema embebido

- IDE Arduino

Aplicación Android

- IDE Android Studio
- Android SDK

3. Diagrama de conexión



4. Implementación

Sistema embebido

Celda de carga

Para su implementación debimos usar un transmisor que se encargue de la interface entre la celda y la placa Arduino, permitiendo leer el peso de manera sencilla. Para utilizar este módulo se debe incluir la biblioteca *HX711.h* y definir 2 pines(DOUT y CLK) por los cuales se comunican de manera serial.

El módulo, internamente, se encarga de la lectura del puente de Wheatstone formado por la celda de carga, convirtiendo la lectura analógica a digital con su conversor A/D interno de 24 bits.

Funciones utilizadas:

HX711 [nombreObjeto](byte PinData, byte PinClock): construye el objeto HX711.

void tare(byte n): establece el peso actual como el peso de tara, *n* indica el número de lecturas que se realizan para obtener la tara.

voidset_scale(floatscale): establece el valor de la escala, que es el factor de conversión para convertir el valor de lectura en un valor con unidades de peso.

floatget_units(byte n): devuelve el valor actual restado del peso de tara y dividido por la escala. Si se especifica un valor de *n*, devuelve el promedio de *n* lecturas.

Motor caja reductora

Para su implementación no se necesitó incluir ninguna biblioteca adicional. Tuvimos que indicar el pin a utilizar (en nuestro caso elegimos el 3), y luego, utilizar las funciones:

digitalWrite(pinMotor, LOW o HIGH): detiene o enciende el motor.

analogWrite(pinMotor, valor): regula la velocidad del motor por PWM. Lo configuramos para tener 5 velocidades.

Servo motor

Para su implementación se debe incluir la biblioteca *Servo.h* y definir el pin a utilizar (en nuestro caso elegimos el 8). Declaramos el objeto con *Servo servoMotor* y utilizamos las funciones:

servoMotor.attach(intpinServo): inicia el servo para trabajar con el pin indicado.

servoMotor.write(intangulo): desplaza el servo al ángulo especificado.

Módulo Bluetooth HC05

Para su implementación se debe incluir la biblioteca *SoftwareSerial.h* declarar el objeto con la instrucción *SoftwareSerialBTSerial(Rx,Tx)*, donde Rx y Tx son los pines donde se conecta el módulo. Utilizamos las siguientes funciones:

BTSerial.available(): comprueba si llega un dato por el puerto bluetooth.

BTSerial.read(): lee el mensaje enviado que se guarda en una variable para luego aplicar la acción correspondiente.

BTSerial.write(): envía el peso medido.

Aplicación Android

Actividades

Actividad inicial para la selección del sistema embebido en un listado de dispositivos bluetooth que están al alcance.

Actividad principal que se encarga de recibir las mediciones que el sistema embebido le envía a través de un hilo asíncrono. Además realiza los cálculos necesarios y envía los mensajes para activar o desactivar los actuadores (motor y servo motor).

Sensores

Para poder manejar los sensores del dispositivo Android se tuvieron que importar las siguientes bibliotecas:

- *android.hardware.Sensor*
- *android.hardware.SensorEventListener*
- *android.hardware.SensorManager*

Con el *SensorManager* indicamos sobre que sensores queremos recibir eventos.

Mediante el evento *onSensorChanged()* estamos constantemente recibiendo los cambios en todos los sensores definidos en el *SensorManager* que son:

Acelerómetro

Se utilizó este sensor para que cuando se detecte un shake comience a funcionar la cinta transportadora.

Proximidad

Se utilizó este sensor para detener la cinta en caso de emergencia.

Orientación

Se utilizó este sensor para que al poner boca abajo el teléfono este vuelque automáticamente el contenido de la balanza.

Bluetooth

Para su implementación se deben importar las bibliotecas:

- *android.bluetooth.BluetoothSocket*
- *android.bluetooth.BluetoothAdapter*
- *android.bluetooth.BluetoothDevice*

5. Funcionamiento

Sistema embebido

El sistema embebido se encarga de enviar constantemente el peso a través de la conexión bluetooth. Además recibe y ejecuta las acciones que debe realizar (encender cinta, detener cinta, regular velocidad y volcar tara) desde el dispositivo Android.

Aplicación Android

La aplicación de Android es la encargada de realizar la conexión bluetooth entre el teléfono y el sistema embebido, el cual debe ser seleccionado de una lista de los dispositivos al alcance.

Luego de la vinculación, la aplicación comienza a recibir el peso medido en el SE y lo muestra en pantalla. El siguiente paso es indicar los parámetros necesarios para el comienzo del proceso automático, que son: la cantidad máxima de elementos y el peso unitario. Al establecer este último, se envía la señal al sistema embebido para que comience a andar la cinta transportadora.

La aplicación, además, es la encargada de realizar los cálculos necesarios para saber si se llegó al número máximo seleccionado y enviar la señal para hacer el volcado de los elementos.

Funciones adicionales:

- Enviar señal de encender cinta al hacer un shake.
- Enviar señal de detener cinta al poner boca abajo el móvil.
- Enviar señal para regular la velocidad de la cinta.
- Enviar señal de volcar tara al tocar el botón correspondiente.

6. Producto terminado

Sistema embebido

Aplicación Android

7. Bibliografía

- Lenguaje de referencia de Arduino. Disponible en Internet:
<https://www.arduino.cc/en/Reference/HomePage>
- Tutorial transmisor de celda de carga HX711, Balanza Digital. Disponible en Internet: http://www.naylampmechatronics.com/blog/25_tutorial-trasmisor-de-celda-de-carga-hx711-ba.html
- Tutorial para que configurar el modulo Bluetooth HC05 en Arduino. [en Formato video]. Disponible en Internet:
https://www.youtube.com/watch?v=XXtf1XtQC_0&t=609s
- Tutorial para que android pueda usar los componentes de Bluetooth. Disponible en internet:
<http://cursoandroidstudio.blogspot.com.ar/2015/10/conexion-bluetooth-android-con-arduino.html>

Código Fuente:

<https://github.com/matiasm91/ProyectoSOA>