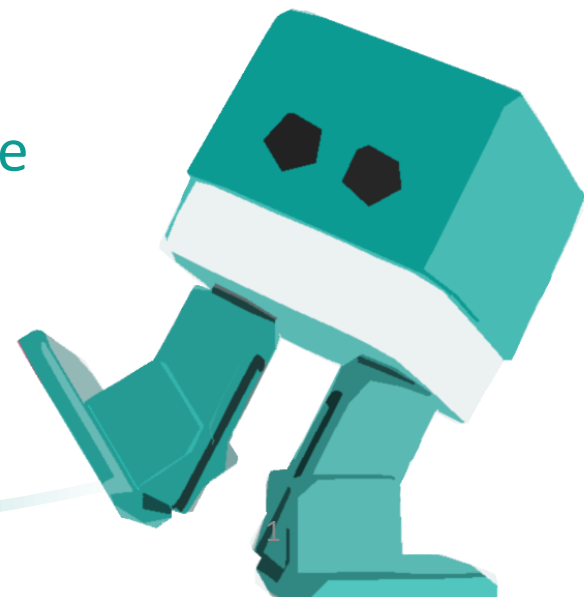




# Banco de Calibración de Zowi

Autor: Marcial Rodríguez

Coordinador: Juan Manuel Enrique





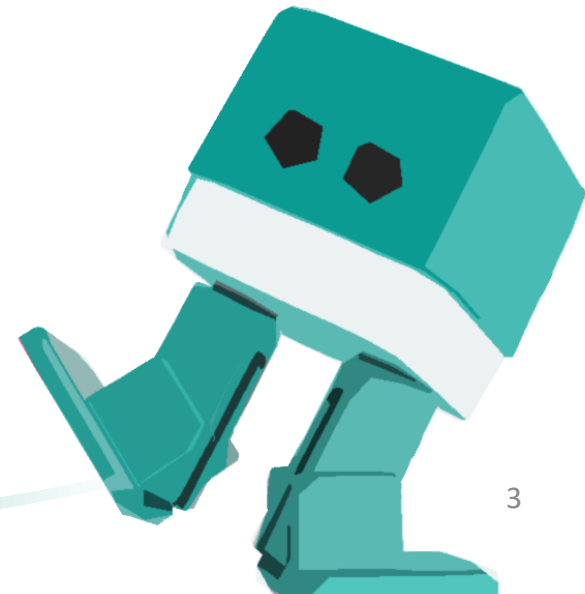
# Índice

- Contexto
- Desarrollo
- Armario Final
- Conclusión



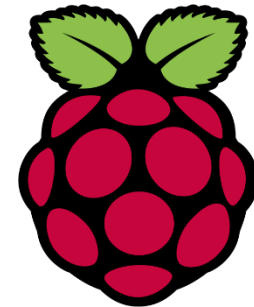
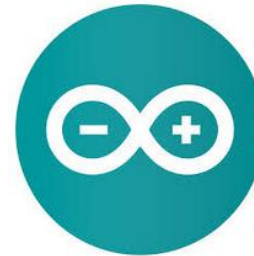
# Contexto

Objetivo, motivo y requisitos del proyecto



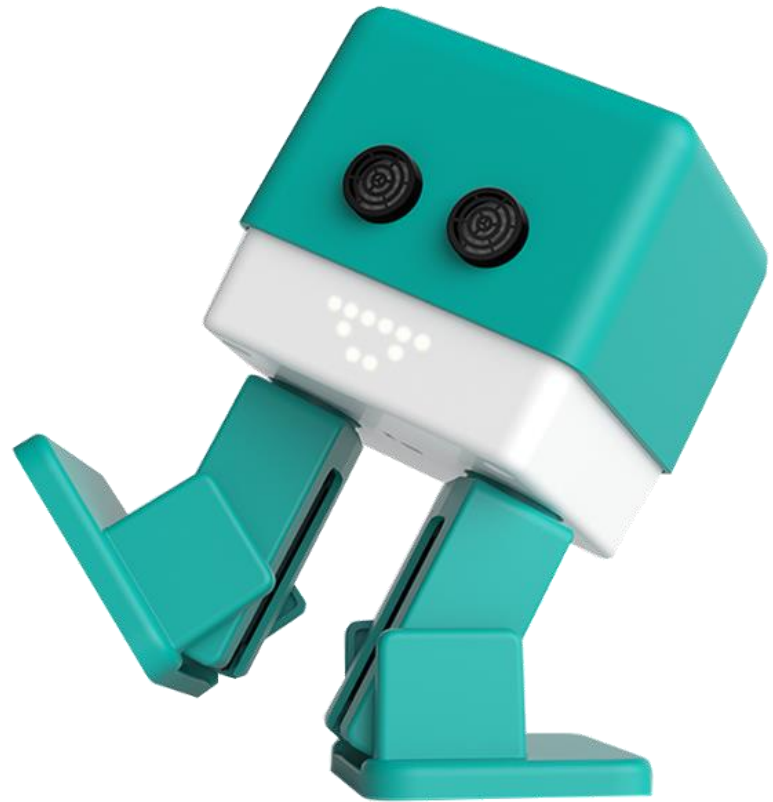
# ¿Qué es este proyecto?

- Es el diseño de un sistema compuesto por:
  - Raspberry
  - Arduino
  - Sensores
  - ...
- Capaz de:
  - Descargar programas en Zowi
  - Medir las posiciones de sus piernas
  - Hacerlo mover sus servos
- Y automatizar parte del proceso de fabricación de Zowi



# Vale, pero... ¿Qué es un Zowi?

- Jugete desarrollado por BQ
- Robot bípedo
- Educativo
- Programable
- Basado en Arduino
- Componentes
  - Servos
  - Ultrasonidos
  - Matriz LED
  - ...



# Vale, pero... ¿Qué es un Zowi?

Zowi es la suma de...

- Diseño derivado del robot Bob
- Librería oscillators
- Aplicaciones y juegos



## BQ y Educación

- Material, cursos y formación de profesorado
- Línea de juguetes educativos: Printbots, kits
- Herramientas: Diwo, bitbloq



# ¿Qué es este proyecto?

## Objetivo

- Diseño y construcción de un sistema parte de la cadena de montaje de Zowi
- Principal función: ajustar de forma automática las posiciones de los servos de Zowi
- Una funcionalidad adicional: descargar del software final en el controlador de Zowi



# ¿Por qué se hace este proyecto?

## Motivo del proyecto

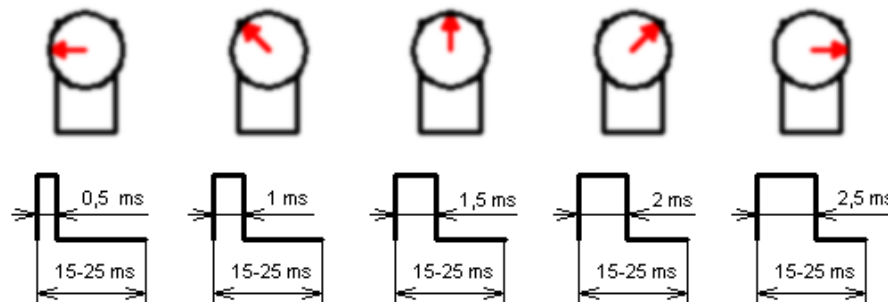
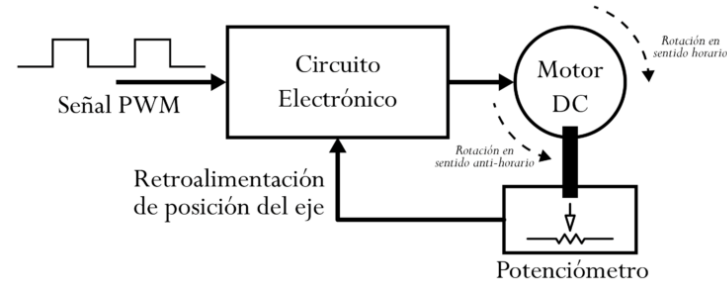
La calibración de los servos resulta ser un problema

- Procedimental
- Específico para cada juguete
- Lento
- ¿Cómo Integrarse en línea de producción?



# Calibración de los servos

- Servos de posición
  - Funcionamiento
- Problema del PWM
- Librería `srvo.h`
- Engranaje a mano en piezas de Zowi
- Fabricación





# Calibración de los servos

## ¿Cómo se calibra un servo?

Tras la instalación del servo y la conexión al controlador

- Definir la posición de partida
- Por ejemplo, asignar un cero
- Ver el desfase frente al valor cero de la librería de Arduino
- Utilizar dicho desfase como “offset” o “trim”



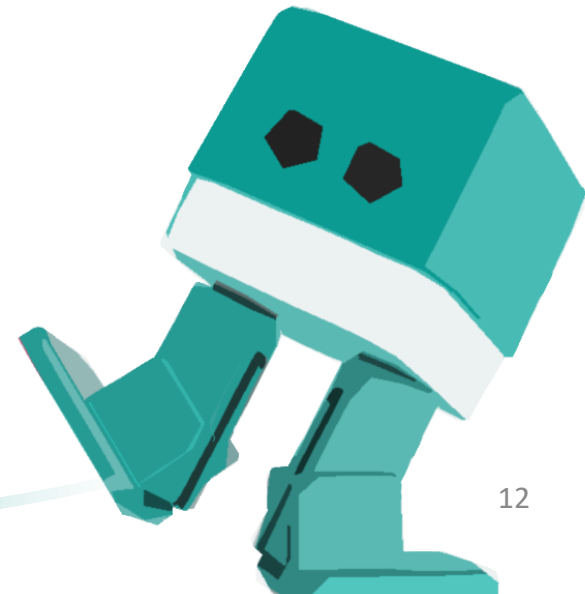
# Requisitos

- Plazo de finalización fijado en 2 meses
- Replicable fácilmente por terceros
- Utilizable por personal no-técnico
- Cadencia aprox: 30-60s
- Fácil instalación



# Desarrollo

## Alternativas y prototipos





## Posibles alternativas

Se realizan algunas pruebas de concepto con diferentes tecnologías

- Visión por computador
- Sensores IMU

Material generado durante el desarrollo disponible en:

- [github.com/uborzz/proyecto-ind](https://github.com/uborzz/proyecto-ind)

# Visión por computador

Se reutiliza sistema montado en célula robotizada

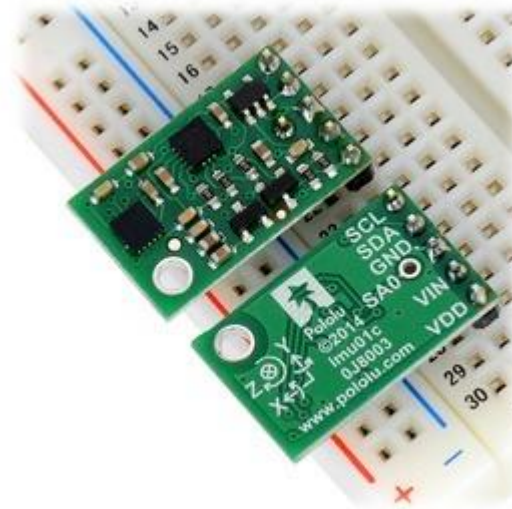
- Cámara Fanuc y iRVision
- Relés 24 – 5 V
- Calibraciones por separado



# Introducción MinIMU-9 v3

Solución elegida de entre varios sensores y librerías

- MinIMU-9 v3 de Pololu con librería MinIMU-9 AHRS Compuesto por acelerómetro, giróscopo y brújula.
- LSM303
- L3GD20H
- 9 DOF
- Precisión 0.1 grados
- I2C (Dirección configurable)



# Solución con IMU

- Posición Zowi
- Anclaje de la IMU
- Calibración – uso (Ejes)
- Elegida frente a la visión por coste, replicabilidad y fácil instalación



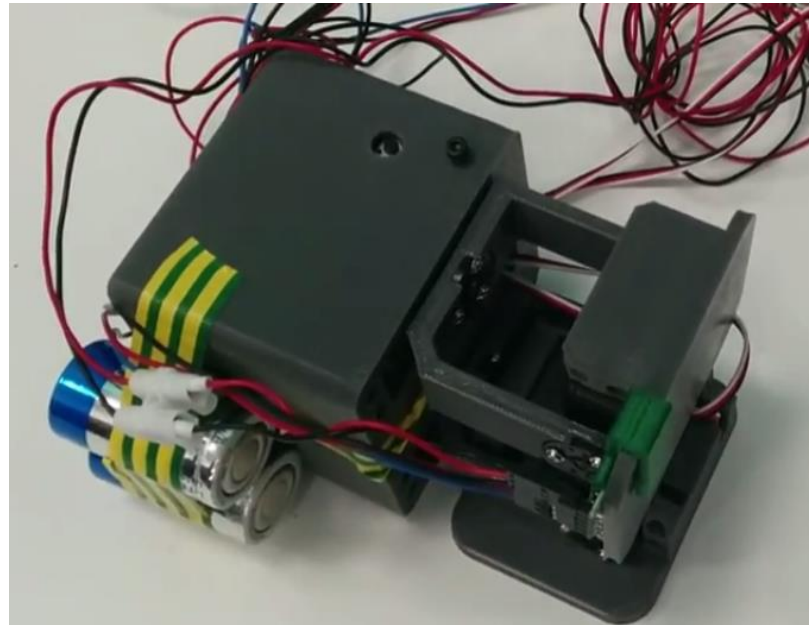
## Procedimiento iterativo:

1. Calibrar IMU
2. Colocar en Zowi
3. Mandar servo a una determinada posición
4. Medir desviación entre posición deseada y real
5. Corregir posición
6. Volver al punto 3 hasta tener la posición deseada
7. Guardar desfase



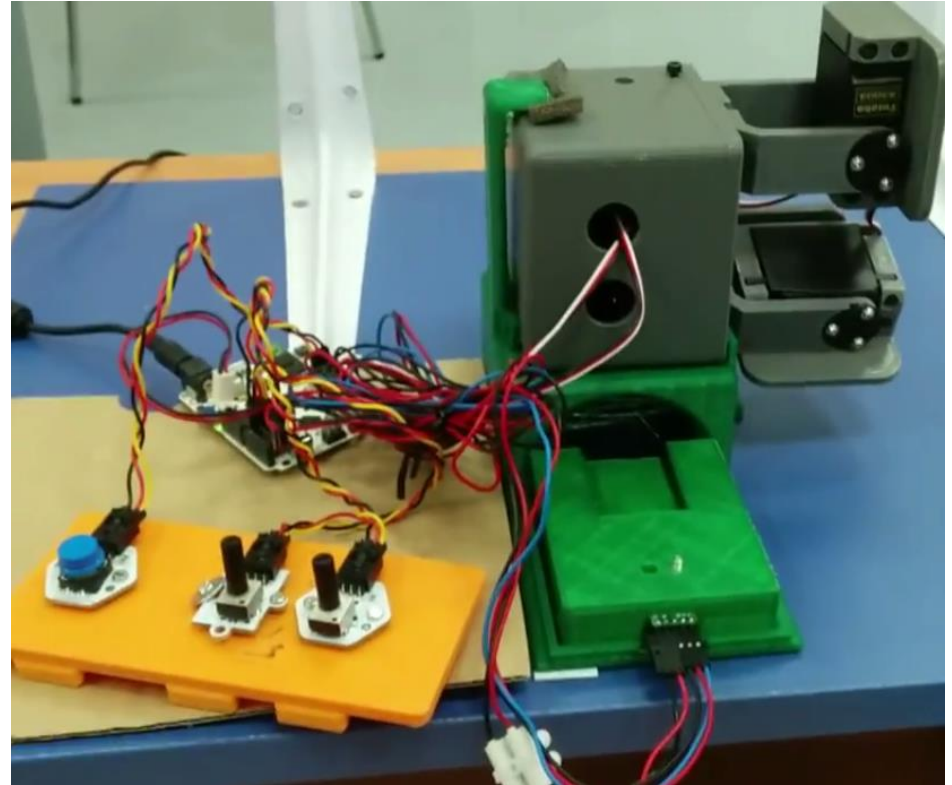
# Prototipo de validación de concepto

- Aprendizaje
- Pruebas diferentes libs y sensor
- Arduino (BQ ZUM) conectado a los servos y IMUs
- Una pierna solamente
- Anclaje con grapa



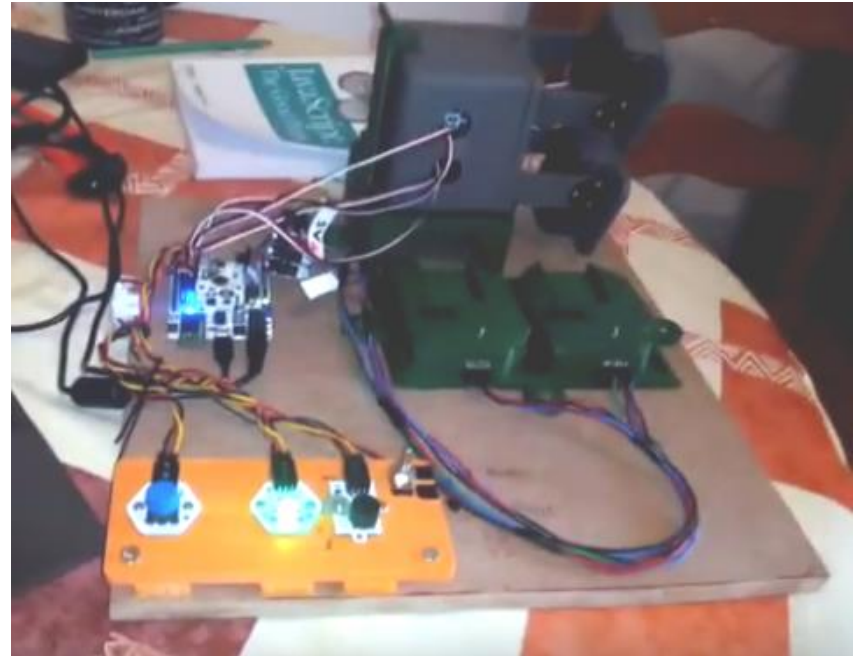
■ ■ ■ ■ Evolución...

- Banco soporte
- Zapato y cajetín
- Paralelismo



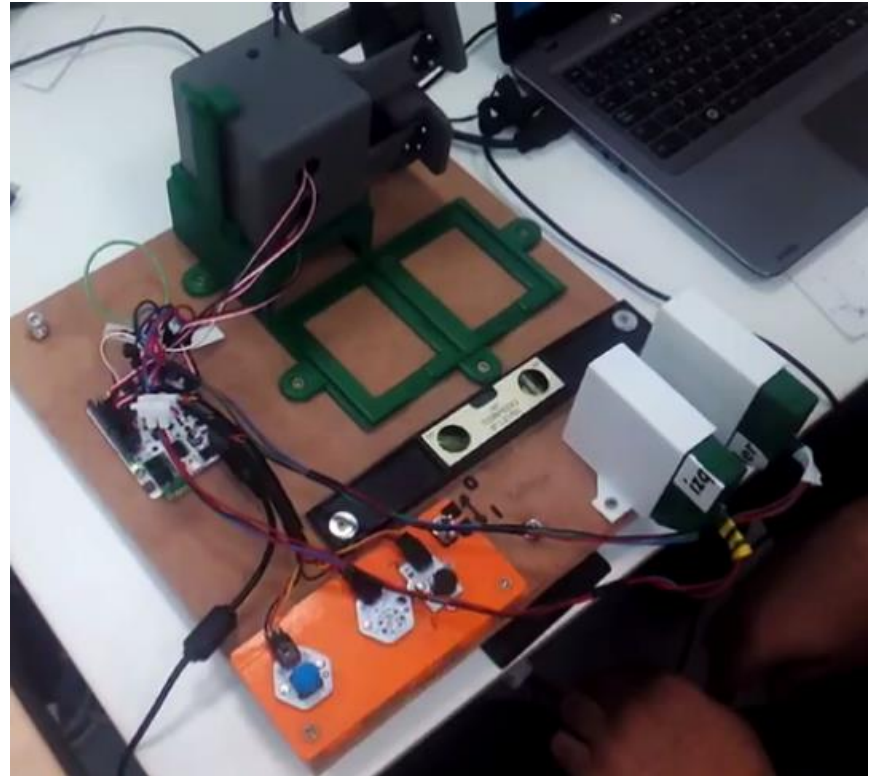
# Evolución...

- Nuevo banco
- 2x IMUs
- Direcciones I2C
- Filtrado
- Más iteraciones
- 0 – 90 deg diferencias



# Evolución...

- Calibración 90 deg
- Mejora calibración inicial
- Nivel y tornillos
- Gestión de errores con los resultados de las mediciones



# Tomando forma...

- Zowi final SSL!
- Freeduino Mega
- Comunicación I2C
  - Wire.h
- Intérprete comandos comunicación
- Cajetines 2 posiciones con poka-yoke
- Endstops en cajetines





# Prototipo final

- Cambio arquitectura
- Conexión USB y problema I2C
- Raspberry como maestro Serie
- Python y migración del intérprete
- Display LCD
- Armario eléctrico y banco de calibración integrados

# Prototipo final

Mejoras software gracias a la Raspberry

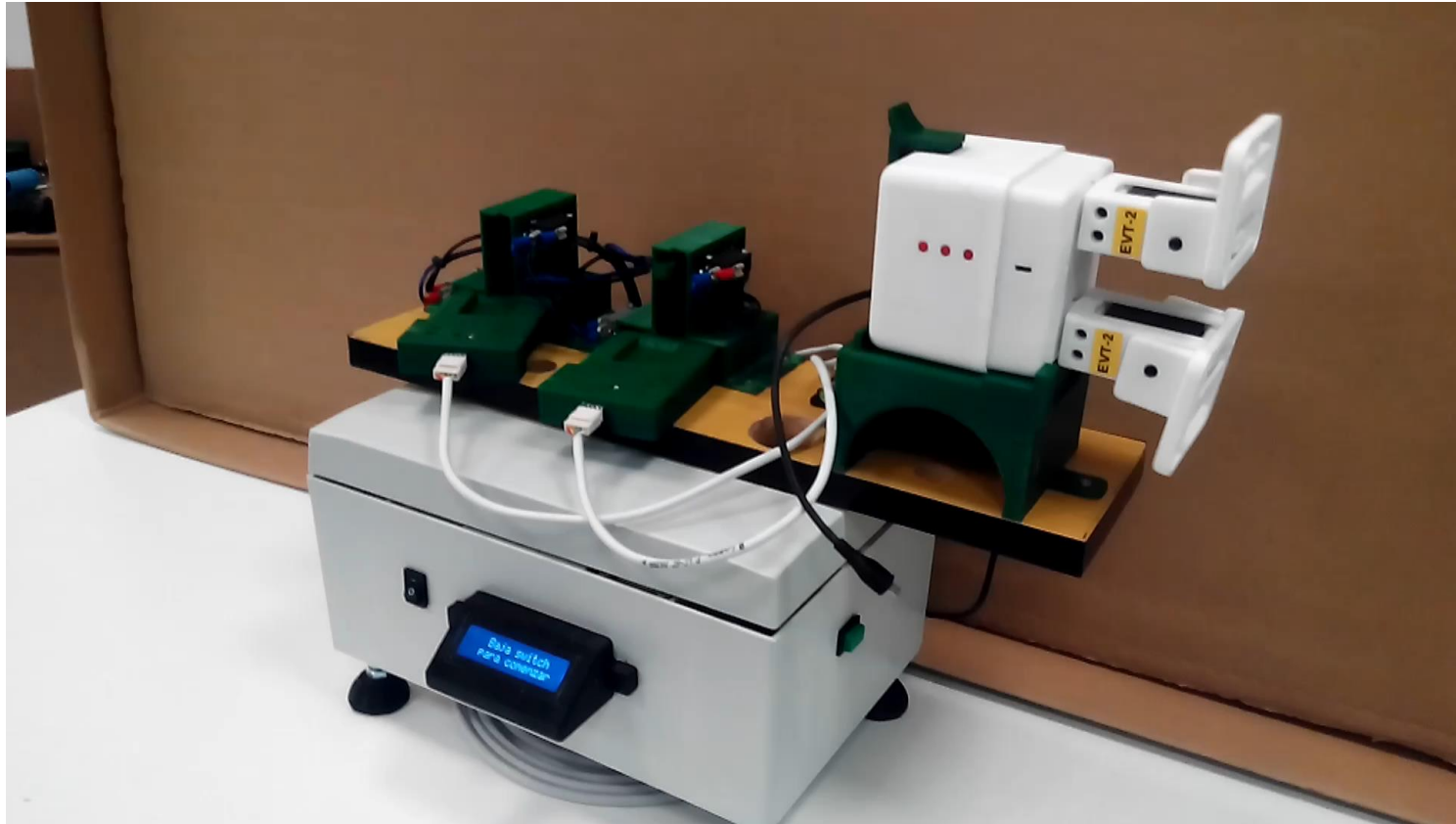
- Acceso por SSH
- MySQL local y remoto
- Acceso por XRDP
- IDE de Arduino
- AVRDUde

Pruebas y validación





# ■ ■ Prototipo final (Vídeo)

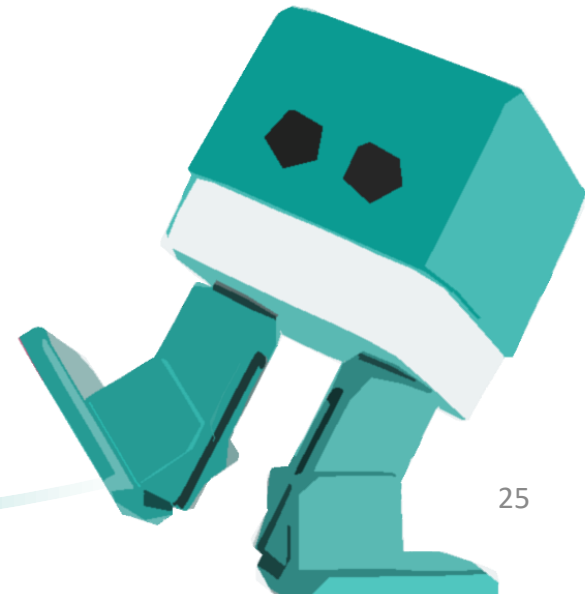






# Armario final

Componentes, software y herramientas



# Armario final

## Cambios solicitados

- Separación Banco-Armario
- Mangueras largas, pulsador
- Conexiones por pasamuros
  - RJ45, USB
- Display altura ojos

## Mejoras adicionales

- Shucko y trafo
- DIM y clemas
- Alimentación pasamuros con interruptor y fusible
- ...

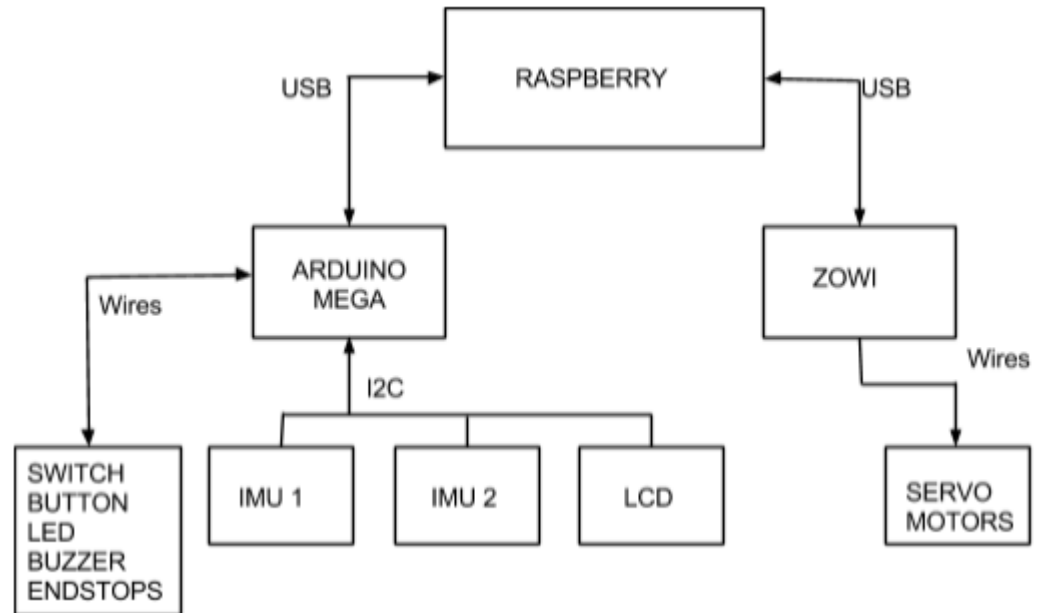


# Armario final



# Esquema de Conexiones

- Lógica cableada
- I2C
- Serial
- Protocolo propio





# Freeduino Mega

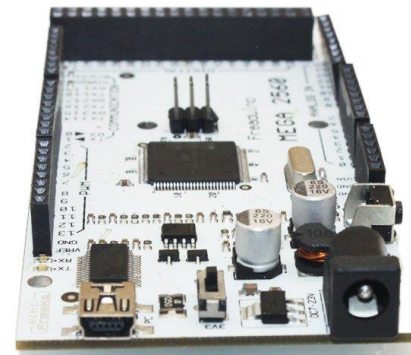
- Cerebro del sistema
- Máquina de estados
- Interacción con el operador

## Hardware

- Leds, buzzer: PWM
- Pulsadores, endstops, botón
  - Pull-ups por software
- Alimentación por Vin

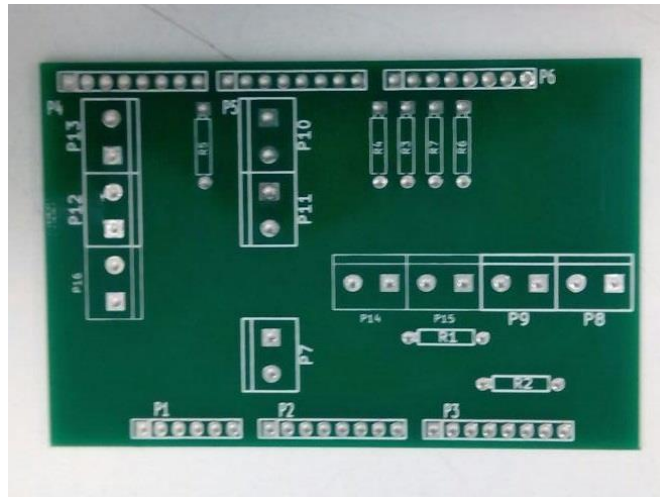
## Software

- IDE Arduino 1.0.6
- Libs: LiquidCrystal, Wire, MinIMU
- Canales serial TTL



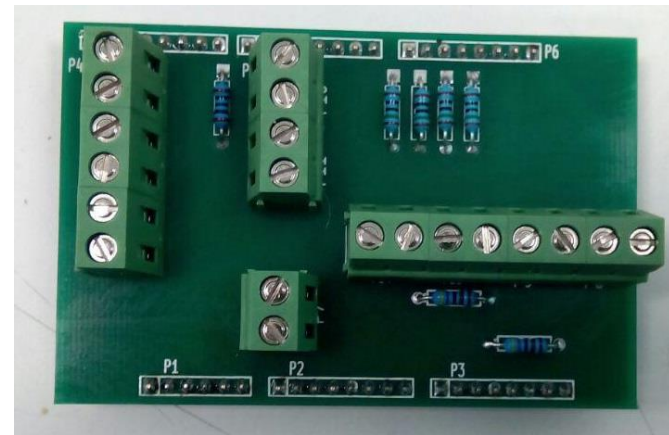
# Shield Mega

- Interfaz salida a bornas
- Resistencias Leds
- Pull-ups I2C



## Prototipado

- KiCAD
- Cyclone, flatcam y CncGcodeController





# Raspberry PI 2

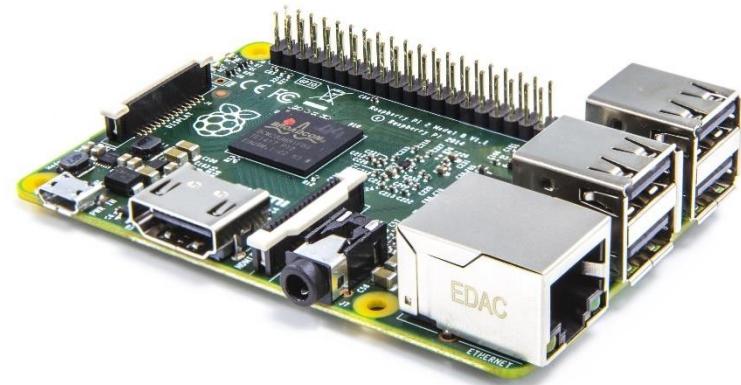
- Interfaz Mega - Zowi
- Programador Zowi
- Salida al exterior

## Software

- SO Raspbian
- Scripts Python y shell

## Hardware

- Trafo
- Alimentación USB Zowi
- Vout a Mega





# Raspberry PI 2

## Configuración

- Current limit: config.txt
- Autologin del usuario
- Script .sh rearranque
- Autostart del .sh
- Protocolo comunicación
- Reglas de los puertos usb
- Bind – unbind
- Programación en Python
- Habilitación SSH
- Instalación XRDP
- MySQL
  - Trazabilidad
  - Detección errores
- IDE Arduino
- AVRdude



# Otros componentes

## Display

- Pantalla LCD
- Backpack de Adafruit
- I2C – LiquidCrystal.h

## Componentes menores

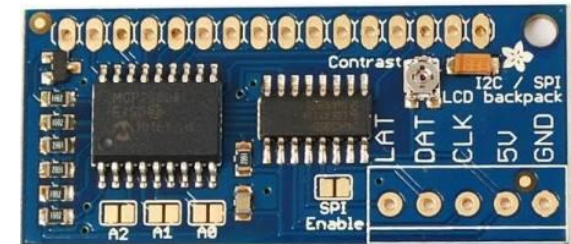
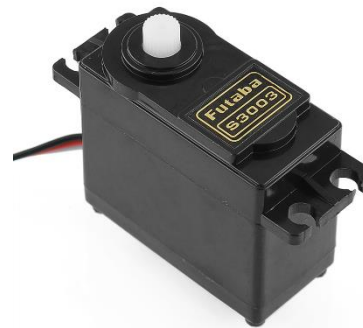
- Buzzer, Led RGB, puntas, clemas, resistencias

## Power Boost

- 5V a 9V aprox.

## Zowi

- Intérprete .ino
- Serial a Raspberry
- Servos
  - Movimiento con srvo.h
- EEPROM.h y I2C\_eeprom.h



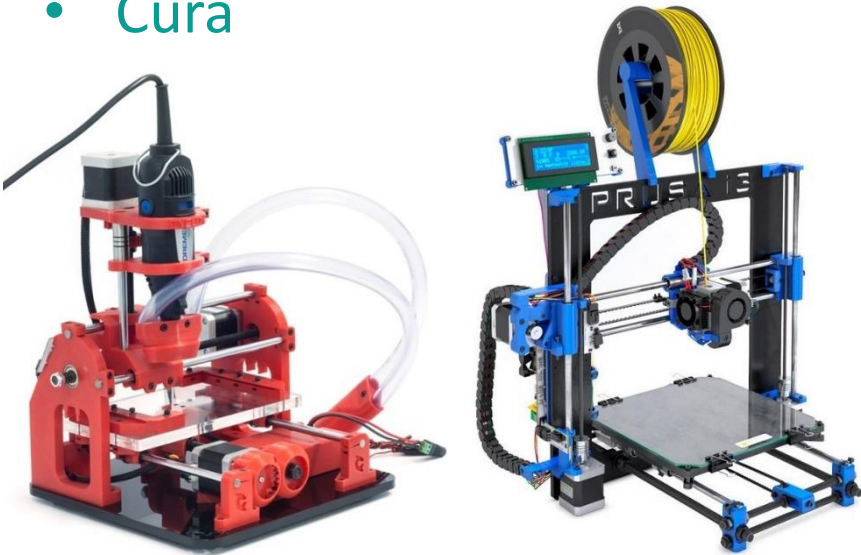
# Tecnologías DIY

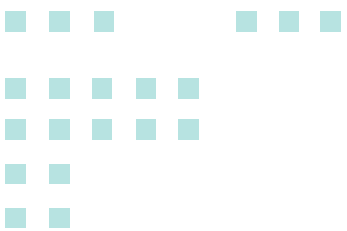
## Piezas impresas

- Prototipado inmediato
- Inventor y Freecad
- Cura

## Aplicación

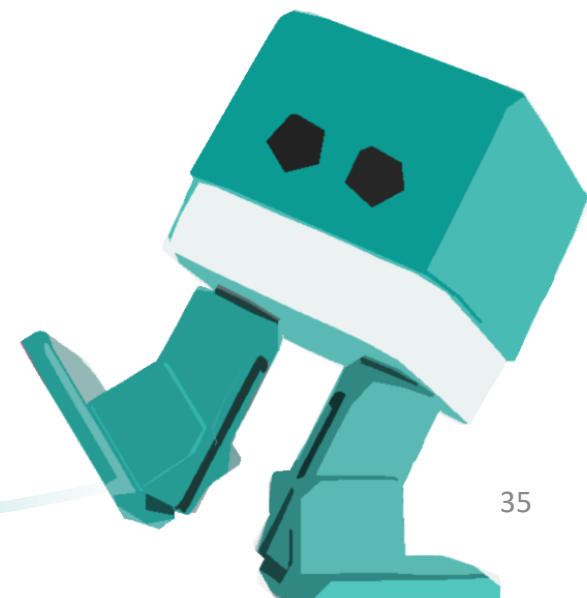
- Bancos soporte prototipo
- Zapatos prototipo
- Anclaje de electrónica a la chapa del armario
- Caja display
- Caja elevador de tensión
- PCB prototipo
- Componentes de Kit

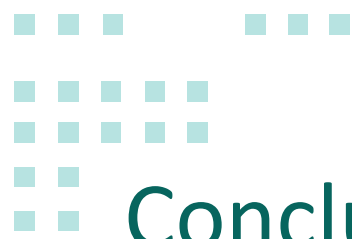




# Conclusión

Resultados y trabajos futuros





# Conclusión

Se afinan los últimos detalles y se envían a Rosti las dos unidades del armario para empezar a funcionar

- Testeo del funcionamiento tras el ensamblado
- Traducciones de mensajes display LCD a inglés
- Redacción de manuales de operación
- Documentación de la máquina de estados del sistema
- Algunas guías cortas para cambio de cableado o instalación
- Backups de las imágenes de las Raspberry
- Elaboración de planos eléctricos



# Resultados

- Instalación en Rosti correctamente
- Últimas mejoras desarrolladas online desde España
- 25000 unidades creadas
- Ventaja 2 armarios
- Prototipo en SAT: Rivas
- Armarios en Marcha: Navarra



# Resultados

## Comparación pre y post mejoras implementadas online

	Pre-mejoras	Post-mejoras
Tiempo calibración medio	81s	50s
Calibraciones a la primera	98.61 %	99.13 %
Calibraciones OK	99.76 %	99.89 %
Apagados “forzados”	63 %	43 %
Recuperaciones vs. restarts	-	35 %

# Resultados

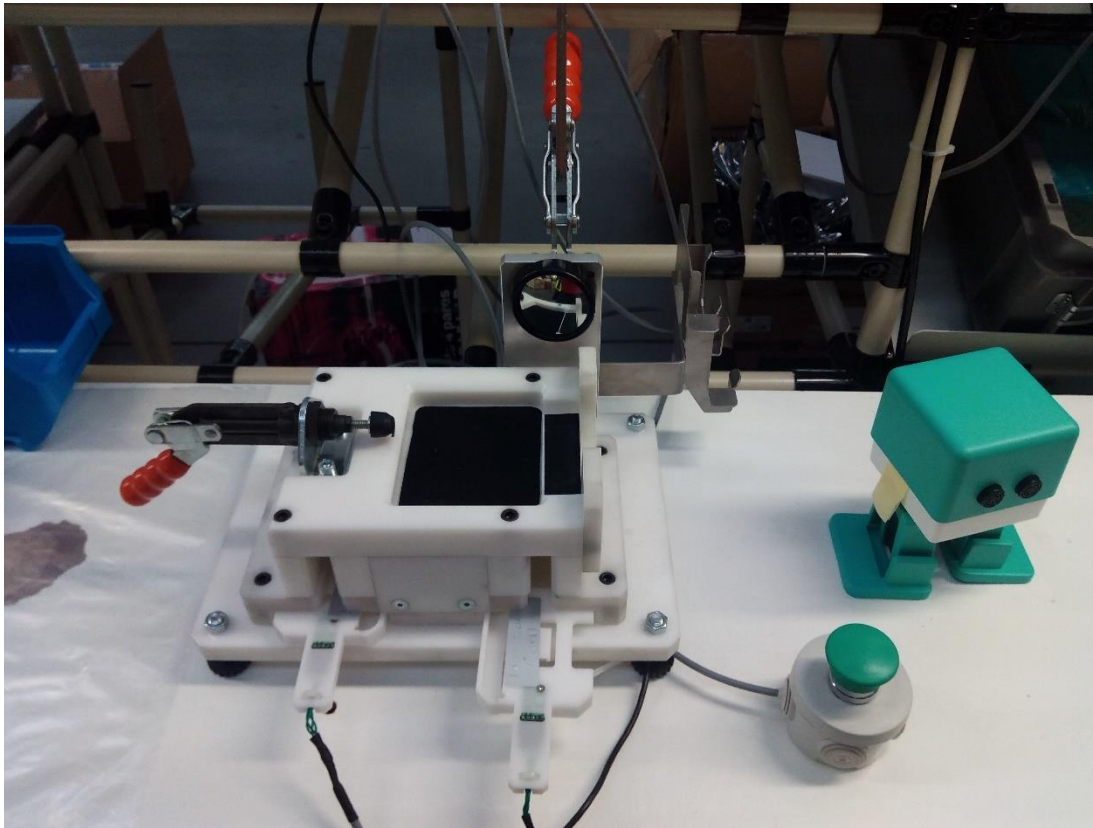
## Fábrica de Rosti y línea de montaje





# Resultados

## Banco y zapatos en fábrica (SSL)







# Trabajos futuros

- Proyecto cerrado
- Proyectos similares con Raspberry y Arduino: Mejoras
  - Implementación de un logger
  - Buen diseño de la base de datos con columna de errores o incidencias, mejoras en la traza del programa
  - Mejorar protocolo comunicación Python-Arduino por serie, implementar requests y forma rápida de definir nuevos comandos, detección de fallos en la interpretación...
  - Emplear un reloj RTC en la Raspberry para no depender de internet y NTP



# Banco de Calibración de Zowi

¡Gracias por su atención!

¿Preguntas?

