# Reunión 17 de diciembre de 2013

## Progresos

* Mejorada comunicación entre Android y Raspberry
  + Añadido un 4 byte -> Numeración de paquete
    - Esto permite descartar los paquetes que han llegado retrasados
* Script al iniciar la Raspberry
  + Al arrancar la Raspberry, carga todo lo necesario para hacer funcionar el robot sin necesidad de interacción.
    - Es decir, al arrancar, sin tocar nada crea los servidores y la red ad hoc.
* Instalación de un sonar
  + OBJETIVO: Prevenir colisiones frontales mediante un frenado de emergencia.
  + En un principio la instalé directamente en la Arduino Due
    - PROBLEMA: Cuando el sonar no tiene ninguna pared cerca bloquea el programa (hasta que lee el tiempo que ha tardado el echo en volver)
    - Solución 1: Añadir un timeout en la lectura.
      * Se reducen los bloqueos, aunque siguen siendo un poco notables.
    - Solución final -> Attiny85v
      * Se delega la lectura del sonar a un microprocesador externo de consumo ultrabajo.
      * Emite una señal (que recibe el Arduino Due) cuando detecta una pared a unos 15cm
      * Ya no se producen bloqueos en el código del Arduino Due
* Cámara
  + En un principio opté por enviar la señal de video a través de netcat “punto a punto”. Es decir, emitir la señal a una IP concreta.
    - ¡Retardos de más de 5 segundos!
  + Alternativa escogida-> Crear un servidor de imágenes. No emito video sino imágenes a baja resolución (320x240) y framerate (15fps)
    - Retardo <0.2s
  + Creado un cliente Android que lee esta señal y la muestra en el móvil.
* Sobre el lenguaje de programación gráfica
  + He tratado de portar Blockly a Android
    - Hay problemas para desplegar el teclado (para introducir números)
* Mejoras menores en el código
  + Corrección de errores
  + Mejorado código y documentación
  + Simplificado el código del Arduino Due

## Progresos en progreso

* Comunicación Raspberry -> Android
  + Se ha creado un hilo tanto en el Android como en Raspberry
    - Android se mantiene siempre a la espera de recibir mensaje de Raspi
    - Raspberry se mantiene a la espera de recibir mensaje de Arduino
      * Para reenviarlo a Android
  + No finalizado el envío de ACK, o señal de frenado de emergencia.
* Servidor DHCP al arrancar la raspberry
  + Se producen fallos arbitrarios: a veces carga, a veces no carga
    - SOLUCIÓN TEMPORAL: IP estática.

## Futuras mejoras

* Crear una comprobación inicial en la aplicación Android que compruebe la conexión con el Robot a través de todos los puertos empleados. Es decir, una especie de ping-pong con el Robot, pero comprobando que sea el robot, y no otro dispositivo, y que todos los puertos están disponibles.
* Mejorar la comunicación Raspberry - Arduino
  + Crear una comprobación inicial de inicio de comunicación
* Leer los sensores y mostrar los valores en Android
* Mejorar presentación Android

## Mejoras posibles

* Programación gráfica basada en Blockly
* Juego de carreras mediante lectura de los sensores
* Grabar recorridos de forma manual
* Siguelineas
* Posibilidad de 2 robots

## Preguntas

* ¿Es aceptable una media de 0.15s de retardo? ¿Se puede mejorar?
* ¿Comunicación bidireccional Attiny – Arduino Due?

## Para la próxima iteración

* Interrupciones
  + Prescindir de Attiny?
* Hacer que funcione SIEMPRE que se encienda, sin necesidad de monitor
* Primera aproximación Blockly – PhoneGap
  + Investigar cómo hacer bloques
* Mejorar (hacer) comunicación Arduino -> Android
* Lectura de sensores IR – SR y envío a Android
* Crear configurador Wifi en Android.
  + Por defecto crea Adhoc, pero q se puedan añadir redes wifi, y conectarse a otros, incluso dar un orden de preferencia de la red a la que conectarse.
* Crear indicadores físicos de estado:
  + Botón físico para hacer red AdHoc
    - ¿En Raspberry, de trompo?
  + Luces que indiquen estados de error
    - Error en comunicación con Arduino
    - Error en cámara
    - Error en WiFi
    - Error en Ad Hoc
    - Ningún dispositivo Android conectado
    - No error (sin bombillas encendidas)
    - Error en las bombillas, LOL
* ¡Hacer funcionar el robot través de Internet!
* S3003 – Servos para mover la cámara.
* Investigar OpenCV
* Poder configurar parámetros desde Android
  + Velocidad máxima lineal y angular.
  + P-I-D para sigue líneas (en el futuro)
  + Distancia de seguridad SR04
  + IP y puerto socket y cámara
  + Parámetros de cámara, resolución y FPS
* Sigue líneas
* Juego de carreras mediante lectura de los sensores
* Grabar recorridos de forma manual
* Posibilidad de 2 robots

**Próxima reunión: 25 de febrero a las 10h**