

Protocolo experimental en Kilobots

Antes de empezar

1. Asegurarse que todos tengan ID:

Iniciar el script *id_debug.hex*. Los kilobots encenderán la luz si tienen un ID $\text{ID} \bmod 10$, i.e. el bot con ID = 1 encenderá del mismo color que el ID 11, 21, 31 y 41. Si no enciende luz, es que no tiene ID.

2. Patas de kilobots rectas:

Utilizar el aparato 3D printed.

3. Comprobar que los kilobots estén calibrados

Iniciar el script *go_straight.hex*. Se considera que un bot está calibrado si pueden ir recto 15 cm, en una anchura de 7.5 cm. Calibrar a los kilobots que se necesite con base en los valores de la tabla:

https://docs.google.com/spreadsheets/d/1ILEAYlaw0cSjxO7ze2P7xQTy3PDm-wXrka6_WbAYmzk/edit#gid=1138819062.

Si estos valores no aseguran ir recto, modificarlos y documentarlo en la tabla.

Pruebas

1. Cuántos mensajes reciben por segundo

script: messages.hex

Protocolo

1. Conectar cable debugging a kilobot, abrir "Serial .." en KiloGUI.
2. Aparecerá el número de mensajes recibidos y los kiloticks.
3. Número de vecinos de 1 a 10.

2. Bots Vistos

script: unique_bots_seen.c

1. Bot emite un mensaje con su ID
2. Bot recibe mensajes con ID.
3. Si es la primera vez que recibe ese ID, guarda ID en vector y sumas 1 a variable de bots vistos.
4. Si recibe un mensaje con el mismo ID, no hace nada.
5. Dependiendo de número de bots vistos, enciende led:
 - 0: no luz
 - 1-3 luz verde
 - 4-6 luz azul
 - 7-10 luz roja
 - 11 en adelante: luz morada

2.1. Bots vistos en Δt

script: Unique_bots_seen_Deltat.c

1. Random walk
2. Bot emite un mensaje con su ID
3. Bot recibe mensajes con ID,
4. Si es la primera vez que recibe ese ID, guardas ID en vector y sumas 1 a variable de bots vistos.
5. Si recibe un mensaje con el mismo ID, no hace nada.

6. En Δt , dependiendo de número de bots vistos, enciende led de un color (luces/intervalos dependientes de Δt) y se detiene.
7. Se toma fotografía del estado final.

3. Desincronización temporal entre bots

4. Random walk tests

script: random_motion.hex

Los vídeos pueden ser vistos aquí:

<https://mega.nz/folder/3GgwDIAD#aVKxpuHiPXFew4ke6cx-Jg>

NOTAS 11 DE NOVIEMBRE 2021

1. Problema de sincronización

- Medir a qué altura los kilobots reciben el infrarrojo de otro kilobot: a qué distancia habría que colgar.
- Para no acumular desfase, probar si hay un desfase constante en los procesadores, y si es así.

Próxima reunión:

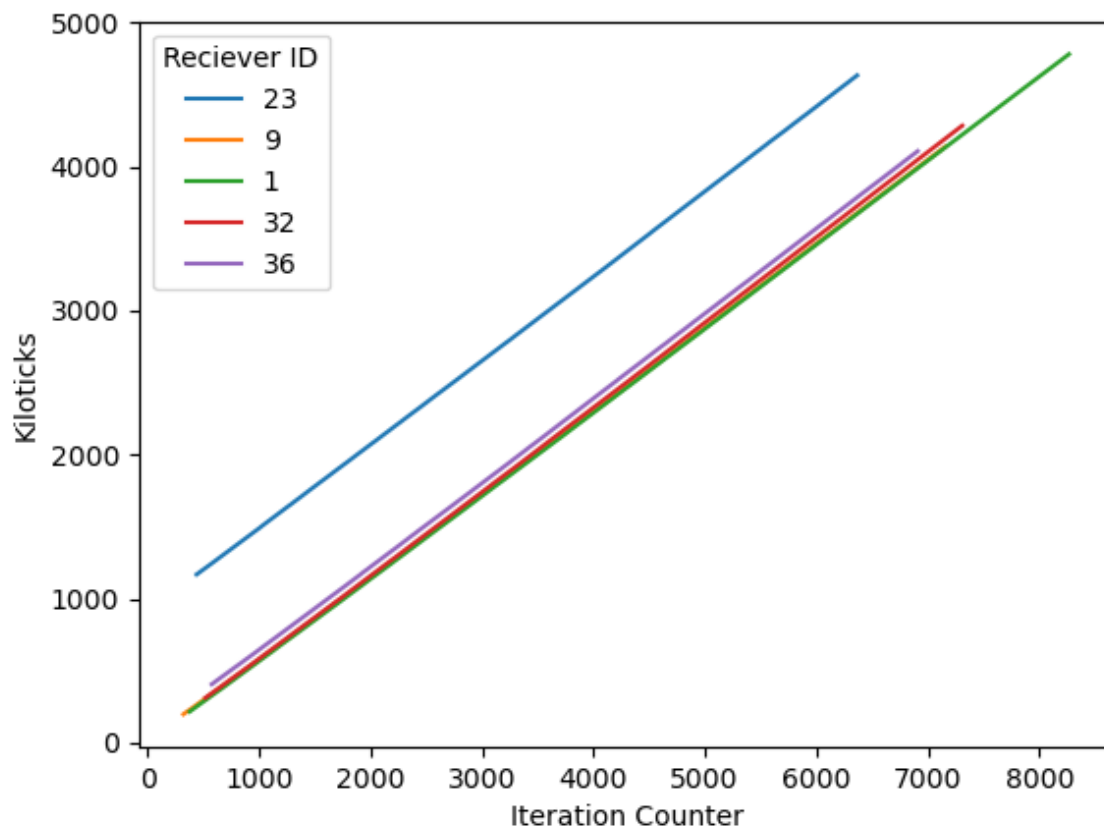
1. Probar con el material de allá.
2. Explicar cómo estoy midiendo los Δt .
3. Ver el efecto de un kilobot que resetee el reloj para todos.
4. Ver el efecto que cuatro kilobots en cuadrantes de medio.
5. Cuántos bots ven en Δt , para varios Δt .
6. Arena pequeña: 20 cm de radio. Hacer vídeos con esta arena y condición PRW-angulos discretos.
7. Modelo de inmóviles con kilombo: mismos vecinos.

Cuánto tiempo pasa en cada loop

Evolución de los kiloticks con las iteraciones del loop: la evolución es lineal con un pendiente de aprox 0.58

$\text{kilo ticks} = 0.58 \cdot \text{iteration}$

Average slope: 0.583 ticks per loop



kiloticks	iterations
100	172.41
500	862
1000	1724.1
5000	8620.6

Pruebas 25/11/2021

Subir el vidrio para ver si aumenta el área de comunicación

CONDICIONES

- Oscuridad
- Tres alturas del vidrio
- 10 kilobots por altura.
- Infrarrojo hacia abajo / hacia arriba

PROTOCOLO

- Poner beacon en el centro, debajo del cristal
- Medir la distancia máxima a la que llega la señal de sincronización
- Anotar la distancia en una tabla.

Tiempo en loops en lugar de en kilo ticks

Elegir la altura de vidrio.

Hacer pruebas de sincronización a partir de loops para $\Delta t \sim 100, 1000, 5000$ ticks

Bots vistos en $\Delta t \sim 100, 1000, 5000$ ticks