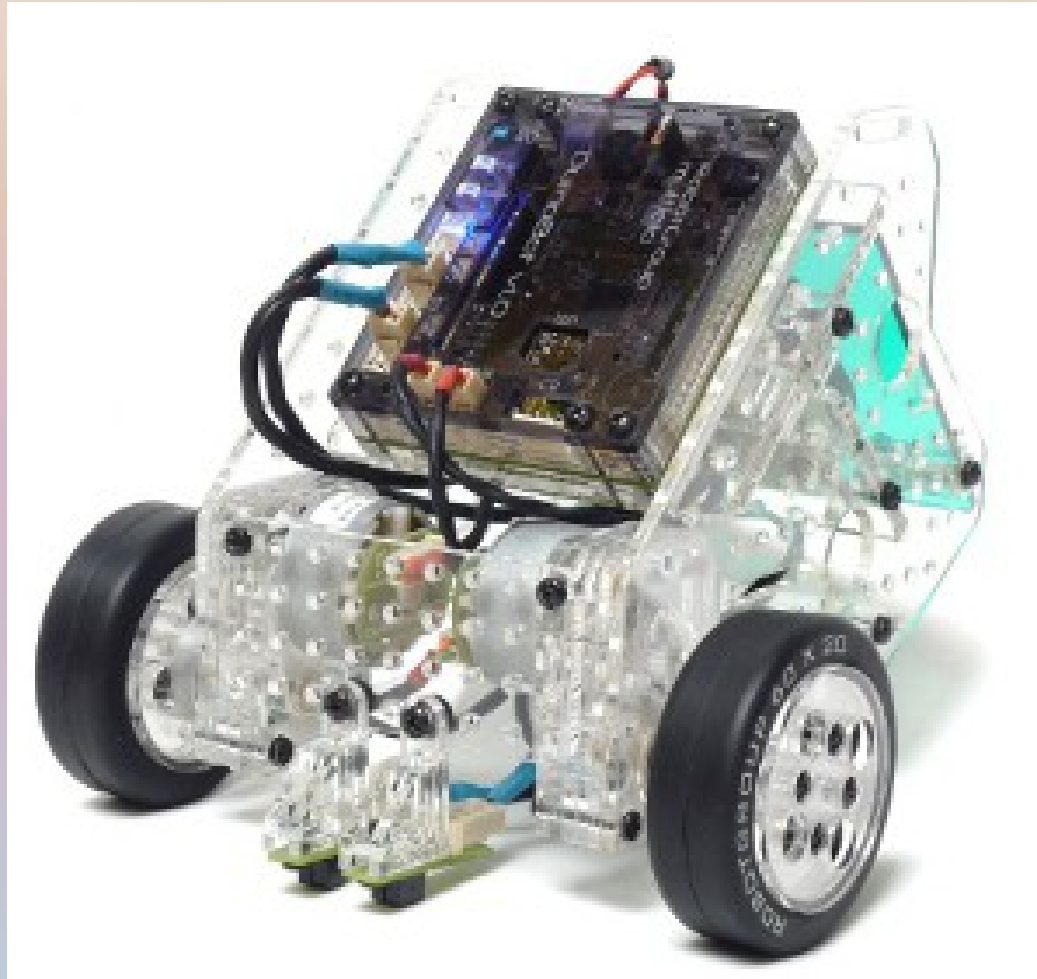


Programación interactiva en Múltiplo N6



Objetivos del trabajo

- Permitir que el robot Múltiple N6 acepte comandos de manera interactiva.
- Construir una API de alto nivel para controlar los robots que permita utilizar la mayor parte del material generado para el proyecto Scribbler.

Características del N6

- Basado en el controlador AVR ATMega32u4.
- Tracción diferencial en dos de sus ruedas.
- Elevador de voltaje a 12V a partir de 3 pilas AA.
- Sensores ultrasónicos, de línea y de carga de batería.
- Leds, switches de power y reset y un buzzer.
- Puerto de comunicaciones USB.
- Software compatible con Arduino.

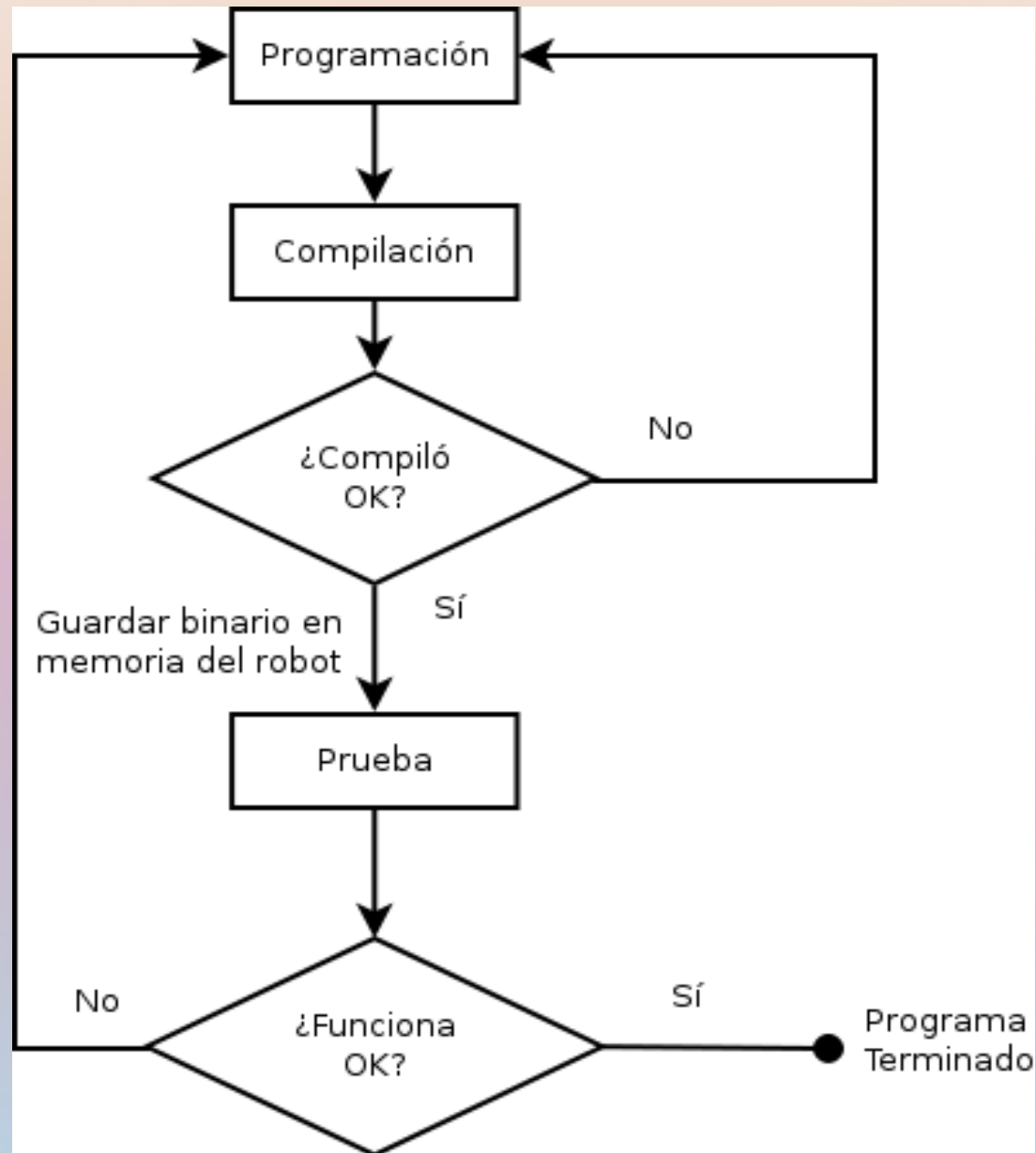
Características del N6

- Como el software es compatible con Arduino, se puede programar utilizando el mismo IDE.
- También soporta Bitlash y Minibloq.
- No tiene forma de comunicarse con un controlador externo (excepto el puerto USB, utilizado para la programación) pero tiene varios puertos libres para agregar componentes como sensores.
- Su licencia RobotGroup-Multiplo Pacifist License (RMPL), basada en MIT License, permite modificar el hardware y el software.

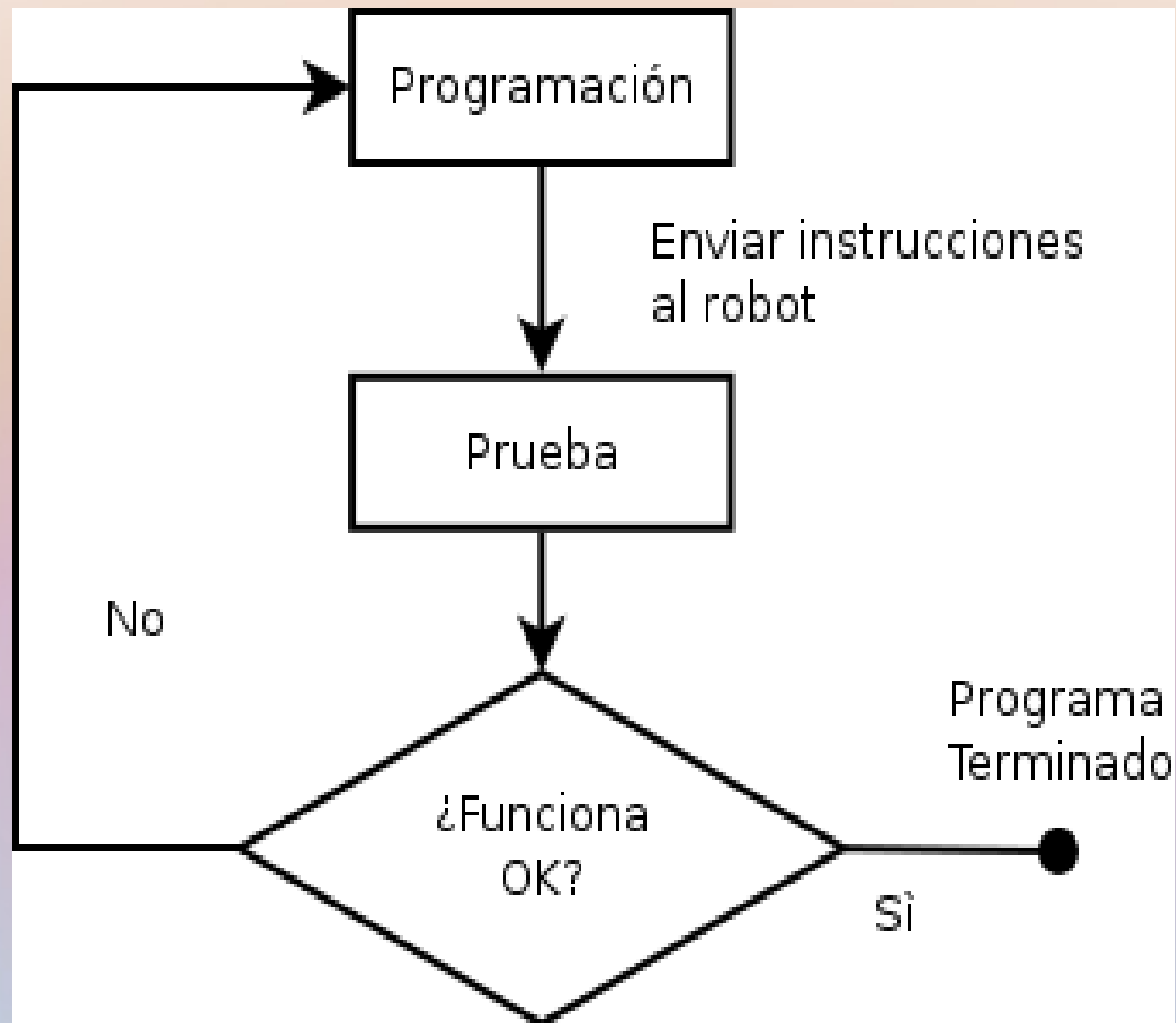
Enseñando a programar con robots

- Se partió de experiencias previas con alumnos de secundario (últimos años), los cuales fueron enfrentados a un manual con ejemplos de código.
- Los alumnos aprendían por prueba y error copiando y modificando el código propuesto.
- Se contaba con robots de un proyecto anterior y se deseaba realizar una experiencia similar.
- Utilizar un lenguaje interpretado y sencillo permite realizar pruebas y correcciones en forma rápida.

Compilado vs Interpretado



Compilado vs Interpretado



Los cambios realizados

Los cambios realizados fueron en 3 frentes

- Hardware
- Firmware
- Software

Los cambios realizados en el Hard

- Básicamente, fue necesario agregar algún método de comunicación para que el robot pueda recibir comandos remotos
- Se optó por la opción de utilizar XBee por su sencillez en cuanto a la integración con Arduino.
- Se trata un emisor/receptor de radio del lado del robot y otro del lado de la PC que controlará al robot.
- El XBee permite conexiones punto a punto y punto a multipunto.

Los cambios realizados en el Firm

- Hablamos de firmware debido a que solamente se graba este programa al robot y ningún otro.
- Este firmware se encarga de interpretar los comandos que le envía la computadora por el protocolo de XBee.
- El firmware se trata de una modificación de Firmata que además permite identificar a que robot se envía cada mensaje.
- El firmware está escrito en el lenguaje en que se programa el AVR y permanece residente.

Los cambios realizados en el Soft

- Los programas para el robot se escriben en Python
- Son interpretados y enviados al robot en forma codificada utilizando PySerial y PyFirmata.
- La programación puede hacerse mediante scripts almacenados en la memoria de la computadora o a través de la consola de Python.
- Esto permite hacer pruebas rápidas que permiten ver si el código escrito hace que el robot se comporte como uno espera.

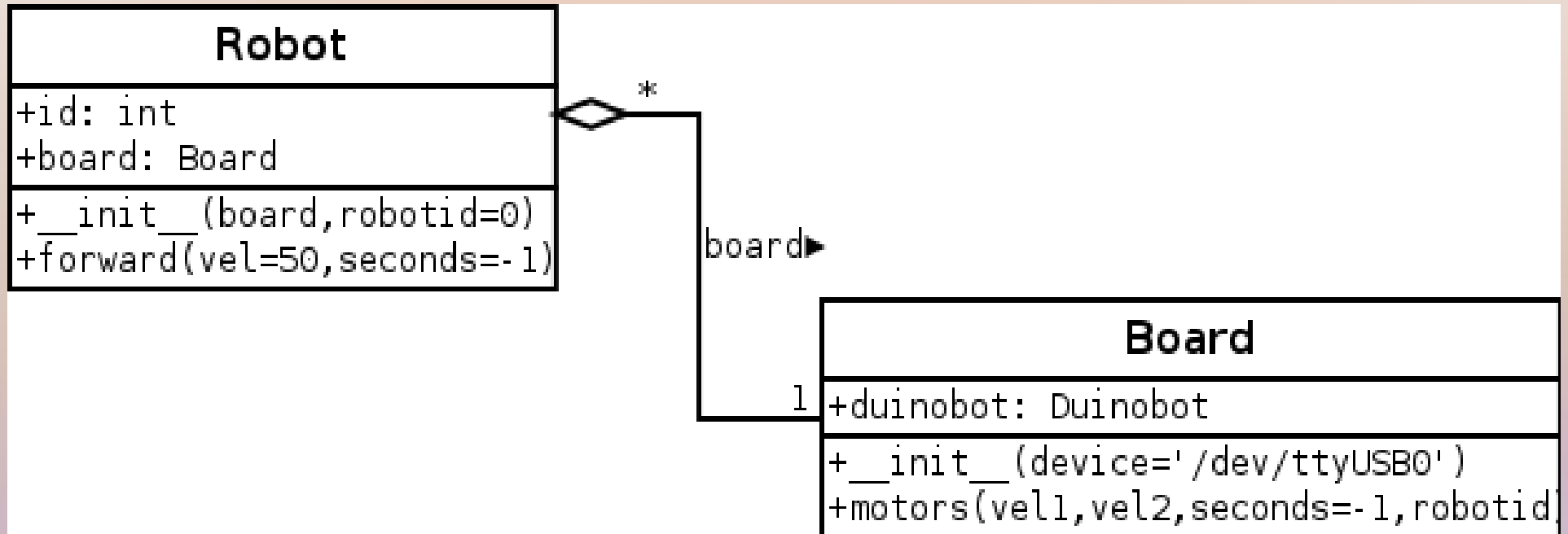
Los cambios realizados en el Soft

- Para que esto sea posible se creó una API en Python que permite interactuar con los robots en forma sencilla.
- La API abstrae el XBee (clase Board) de más bajo nivel que permite interactuar con varios robots a la vez.
- Los mensajes de esta clase en general requieren cierta especificidad, por ejemplo para identificar a que robot va dirigido el mensaje.
- Los mensajes son enviados haciendo referencia a el objeto Board que identifica un dongle particular.
- Por ejemplo, para que el robot con id=1 avance 10 segundos:
 - `b.motors(50, 50, 10, 1)`

Los cambios realizados en el Soft

- También abstrae a un robot (clase Robot) que permite crear una instancia de un robot y enviarle mensajes.
- Es necesario, cuando se crea el objeto robot, asociarlo a un dongle.
- Por ejemplo, para hacer que el robot avance 10 segundos:
 - `r.forward(10)`
- Como cada robot conoce su identificador, no es necesario enviarlo.

Los cambios realizados en el Soft



Integración con el proyecto Lihuen GNU/Linux

- Para integrar este trabajo con el proyecto Lihuen fue necesario crear un paquete instalable (.deb) que una vez agregado a los repositorios, pudiera instalarse con las herramientas provistas por la distribución.
- La última versión de este paquete es `robot_0.09_all.deb`.
- Además existe un proyecto del mismo grupo de trabajo para poner en práctica la enseñanza de programación con Python utilizando robots personales.

Trabajo a futuro

- Investigar más a fondo el tema de interacción entre robots.
- Integrar la API de Python en Minibloq
- Hacer pruebas con sistemas económicos de visión por computador utilizando webcams y librerías como OpenCV.