

Krotic V.E.



Instituto Tecnológico de Costa Rica
Escuela de computación
Centro de Investigaciones en Computación

KROTIC: Kit de Robótica Costarricense Versión Extendida

Informe final elaborado por:

Milton Villegas Lemus

Cartago, Costa Rica
Noviembre 27, 2017

Tabla de Contenido

1	KROTIC Kit de Robótica Costarricense versión extendida.....	8
2	Introducción	10
2.1	Objetivo General	11
2.2	Objetivos Específicos.....	11
3	Marco Teórico	12
3.1	Antecedentes	12
3.2	Robots en el aula.....	12
3.3	Arquitectura de Software para Robots	13
3.4	Máquinas Virtuales	15
3.5	Unidad de Control Microprogramada.....	15
3.6	Descripción de los principales dispositivos evaluados:.....	17
3.6.1	WiFi-Shield:	17
3.6.2	Módulo RF24:	18
4	Metodología	18
4.1	Software	20
4.2	Hardware.....	21
5	Resultados	22
5.1	Programador	22
5.2	Pruebas de duración y consumo durante quemado de Sketch:	28
5.2.1	Primera Iteración:.....	29
5.2.2	Segunda Iteración:.....	31
5.3	Robot Krotic.....	35
5.4	Módulo de Comunicación Inalámbrica	56
5.5	Cámaras.....	71

5.6	Plan Piloto para Escuelas.....	75
6	Discusión y Conclusiones.....	80
6.1	Quemador (Tricorder).....	80
6.2	Robot KROTIC.....	80
6.3	Comunicación Inalámbrica.....	81
6.3.1	Prueba con opción alternativa de almacenamiento:.....	81
6.4	Plan Piloto para Escuelas.....	82
7	Recomendaciones.....	83
7.1	Programador.....	83
7.2	Set de Instrucciones.....	83
7.3	Krotic (Hardware).....	84
7.4	Plan Piloto para Escuelas.....	84
8	Referencias.....	85

Índice de Cuadros

Tabla 1 Set de Instrucciones del Krotic.....	24
Tabla 2 Códigos de programa de prueba.....	29
Tabla 3 Mediciones Batería A, iteración 1	30
Tabla 4 Mediciones Batería B, iteración 1	30
Tabla 5 Medición Batería A, iteración 2	31
Tabla 6 Medición Batería B, iteración 2	32
Tabla 7 Multímetro 1 usado en las mediciones de tensión.....	36
Tabla 8 Multímetro 2 usado en las mediciones de tensión.....	36
Tabla 9 Mediciones con batería recargable 9V, primera ronda.....	37
Tabla 10 Mediciones con batería recargable 9V, segunda ronda	37
Tabla 11 Mediciones con batería convencional de 9V, primera ronda	38
Tabla 12 Mediciones con batería convencional 9V, segunda ronda	38
Tabla 13 Mediciones batería recargable AA, primera ronda.....	39
Tabla 14 Mediciones batería recargable AA, segunda ronda.....	39
Tabla 15 Mediciones batería convencional AA, primera ronda.....	40
Tabla 16 Mediciones batería convencional AA, segunda ronda	40
Tabla 17 Mediciones de consumo en el Krotic batería recargable AA, algoritmo Subsumption seguir luz, primera ronda.....	42
Tabla 18 Mediciones de consumo en el Krotic batería recargable AA, algoritmo Subsumption seguir luz, segunda ronda.....	43
Tabla 19 Mediciones de consumo en el Krotic batería convencional AA, algoritmo Subsumption seguir luz, primera ronda.....	44
Tabla 20 Mediciones de consumo en el Krotic batería convencional AA, algoritmo Subsumption seguir luz, segunda ronda.....	45
Tabla 21 Mediciones de consumo en el Krotic batería recargable 9V, algoritmo Subsumption seguir luz, primera ronda	46

Tabla 22 Mediciones de consumo en el Krotic batería recargable 9V, algoritmo Subsumption seguir luz, segunda ronda.....	47
Tabla 23 Mediciones de consumo en el Krotic batería convencional 9V, algoritmo Subsumption seguir luz, primera ronda.....	48
Tabla 24 Mediciones de consumo en el Krotic batería convencional 9V, algoritmo Subsumption seguir luz, segunda ronda.....	49
Tabla 25 Comprobación de consumo máximo del servo comercial.....	51
Tabla 26 Mediciones utilizando WIFI-Shield a 2m del servidor	57
Tabla 27 Mediciones utilizando WIFI-Shield a 11.85m del servidor	57
Tabla 28 Mediciones utilizando módulo RF24 a 2m del servidor	58
Tabla 29 Mediciones utilizando módulo RF24 a 11.85m del servidor	58
Tabla 30 Tiempos de transferencia de patrón 1.....	62
Tabla 31 Tiempos de Transferencia de patrón 2	62
Tabla 32 Tiempos de transferencia de patrón 3.....	63
Tabla 33 Segunda prueba con patrón 1	63
Tabla 34 Segunda prueba con patrón 2	64
Tabla 35 Segunda prueba con patrón 3	64
Tabla 36 Tercera prueba utilizando patrón 1.....	64
Tabla 37 Tercera prueba utilizando patrón 2.....	65
Tabla 38 Tercera prueba utilizando patrón 3.....	65
Tabla 39 Tiempos de transferencia de patrón 1 a 2 metros del punto de acceso.....	66
Tabla 40 Tiempo de transferencia de patrón 2, a 2 metros del punto de acceso	66
Tabla 41 Tiempo de transferencia de patrón 3, a 2 metros del punto de acceso	66
Tabla 42 Tiempos de transferencia de patrón 1, a 11.85 metros del servidor	67
Tabla 43 Tiempo de transferencia de patrón 2 a 11.85 metros del servidor.....	68
Tabla 44 Tiempo de transferencia de patrón 3 a 11.85 metros del servidor.....	68

Tabla 45 Mediciones de autonomía utilizando módulo RF24.....	70
Tabla 46 Mediciones de autonomía utilizando módulo WiFi-Shied.....	70
Tabla 47 Mediciones de consumo de corriente de Raspberry Pi B v1 en diferentes condiciones. Todos los valores se presentan en Amperios (A).....	71

Índice de Gráficos

Gráfico 1 Comparación de Fuentes de Alimentación, 9V.....	41
Gráfico 2 Comparación Fuentes de Alimentación, AA 8,4V.....	41
Gráfico 3 Comparación Corriente Diferentes Fuentes.....	50
Gráfico 4 Comparación Consumo Diferentes Corrientes.....	51
Gráfico 5 Comparación Módulos de Transmisión Inalámbrica 2m.....	59
Gráfico 6 Comparación Módulos Transmisión Inalámbricos 11.85 cm.....	60
Gráfico 7 Tiempos Promedios de Tres Patrones por Microcontrolador 2m.....	67
Gráfico 8 Tiempos Promedios de Tres Patrones por Microcontrolador 11.85 m.....	69
Gráfico 9 Corriente Demandada por la Cámara con diferentes programadores.....	73

Índice de figuras

Figura 1 Funcionamiento de una Unidad de Control Microprogramada según Stallings (Stallings, 2013).	16
Figura 2 Imagen de módulo Wifi-Shield	17
Figura 3 Imagen de módulo RF24	18
Figura 4 Adaptación Metodología Agile aplicada al proyecto.....	20
Figura 5 Diagrama esquemático del primer diseño del quemador incluyendo el RaspberryPi.....	33
Figura 6 Diagrama electrónico esquemático del quemador con el que se construyeron los prototipos.....	34
Figura 7 Tricorder conectado para transferir programa al microcontrolador del Robot	34
Figura 8 Robot Krotic en el proceso de grabación de E2PROM	52
Figura 9 Programa de seguimiento de luz.....	54
Figura 10 Programa sin decisión.....	54
Figura 11 Programa con decisión.....	55
Figura 12 Cámara 1 probada Raspberry Pi, vista de frente.....	73
Figura 13 Cámara Vimicro, vista del lente	74
Figura 14 Cámara Vimicro, vista del circuito	74

1 KROTIC Kit de Robótica Costarricense versión extendida

Resumen

Krotic, Kit de robótica costarricense, es un producto del proyecto de investigación del mismo nombre, inscrito en la VIE 2008-2010. Dicho proyecto logró construir un kit de robótica de \$75 dólares con fines educativos.

Finalizado el proyecto en la VIE, se continuó trabajando en el proyecto, gracias al apoyo de CIC, llegando a un diseño atractivo y funcional que permitía la adaptación del robot a diferentes entornos de aprendizaje.

En este proyecto, la versión extendida de Krotic, se plantearon como principales objetivos: construir un módulo para programar los robots sin utilizar computador convencional; construir un módulo con cámara; construir un módulo de comunicación inalámbrica; implantar una arquitectura Subsunción, evaluándola principalmente en consumo de energía; elaborar una propuesta de aplicación de los Krotic en el aula para tres escuelas como plan piloto.

En el diseño y construcción tanto de los módulos como del programador se evaluaron opciones de mercado y plataformas de desarrollo. También se midieron las variables, en especial de consumo, variable de estado en este tipo de sistemas.

Con el diseño y la construcción de los circuitos propuestos finalizados, se construyeron diez prototipos de los robots para usarlos en conjunto con los módulos de programación en un esquema piloto. Del dispositivo programador se construyeron dos. Se efectuaron mediciones del desempeño de los algoritmos en cada uno de los dispositivos.

Se logró cada uno de los objetivos del proyecto, como conclusión principal, se estableció que para ejecutar de mejor manera la arquitectura subsunción es mejor dedicar un procesador a ello, para que la programación del comportamiento pueda contar con recursos suficientes para ejecutarse satisfactoriamente.

Abstract

Krotic Costa Rican Robotic Kit is an outcome of a VIE research project during 2008-2010. The robotic kit was constructed for educational purposes with a cost of \$75.

The structural analysis was continued thanks to the support of the CIC and de Computer Science School, achieving a very nice, attractive and functional design allowing the robot be used in different kind of learning projects.

The main goals of Extended Version of Krotic project are: construct a programmer device for the robot without using regular computer; a camera module; a wireless communication module; program and evaluation of a Subsumption architecture measuring energy consumption; propose a didactical plan to use in 3 elementary schools as a pilot test.

During the design and construction process development kits and circuits available were assessed. Some variable were measured specially consumption, which is a state variable in these systems.

Once the circuits were finished, ten robotic kits were constructed in order to use them with the programmer in a pilot plan in the schools. Two programming modules were constructed. The time response for every algorithm in the different device constructed were measured.

The project achieved all the objectives set, the main conclusion states that it is better have a dedicated microcontroller for the execution of the Subsumption subroutines and for the rest of the robot jobs have another microcontroller.

Palabras clave:

Robots personales, educación, construccionismo, arquitectura de subsunción.

Keywords Personal robots, education, constructionism, Subsumption architecture.

2 Introducción

Krotic un Kit de robótica que fue desarrollado en un proyecto de investigación inscrito en la VIE 2008-2010. Como uno de los productos se logró construir el kit de robótica con fines educativos a un precio de \$75 dólares. El Kit está compuesto por un conjunto de partes de ensamblaje rápido, una tarjeta principal con el procesador del robot, sensores, una tarjeta para prototipos para conectar diferentes tipos de sensores así como electrónica complementaria para ellos, fuente interna y conexión a fuente externa, un par de servomotores.

El diseño de la parte estructural se continuó después de finalizado el proyecto en la VIE, gracias al apoyo de CIC, llegando a un diseño atractivo que permite la adaptación del robot a diferentes entornos de aprendizaje.

Para que un robot tenga aplicabilidad en el aula con el objeto de interesar y estimular en los estudiantes el aprendizaje de conceptos de ciencias básicas y tecnología, debe contar con funcionalidades de forma que responda a una buena parte de las iniciativas de los estudiantes. El módulo para uso de una cámara se muestra como uno de las funcionalidades más atractivas para los estudiantes. Solamente hay que considerar que hoy día la gran mayoría de teléfonos celulares posee cámara. Otro módulo que se plantea útil es el de la comunicación inalámbrica, este módulo presenta una gran variedad de situaciones con la posibilidad de transferencia de datos entre robots y posibilidades de escenarios para diferentes proyectos de los estudiantes. Otro circuito fundamental es el de programación del robot, para ello se estimó que era necesario que no se tuviera necesidad de un equipo convencional, ya que por experiencia del laboratorio LuTec en algunas escuelas rurales no todos los estudiantes tienen acceso a dicha tecnología o si la hay no necesariamente cuentan con versiones de sistema operativo que lo puedan hacer funcional. La solución para este problema es construir un dispositivo para programar el robot de bajo costo adoptando un enfoque por objetivos [1] (Villegas-Lemus, 2010). No menos importante es la forma de cómo será el uso de los robots en el aula, que de acuerdo con la literatura se ha definido de importancia estratégica como lo muestra proyecto de escala Europea TERCoP [7] (Dimitris Alimisis, 2006), es importante mencionar que la estrategia de abordaje para el uso de tecnología fue discutida, planteada y puesta a prueba con colegas de la Universidad de Sao Paulo en Brasil.

Aspectos de diseño, es decir tales como el factor de forma del robot o esquema de ensamblaje, tanto del robot como de su dispositivo de programación, son parte del desarrollo integral de esta propuesta, pero están fuera del alcance de esta investigación.

Para otorgar al Krotic con más funcionalidades de forma que sea versátil su aplicación en la didáctica del aula se planteó a la VIE una extensión al proyecto Krotic.

El objetivo general y los objetivos específicos del proyecto se plantean a continuación:

2.1 Objetivo General

Diseñar, Construir la versión extendida del Kit de Robótica Costarricense, su dispositivo de programación y elaborar un plan de piloto de aplicación en 3 escuelas.

2.2 Objetivos Específicos

- Diseñar evaluar y Construir el Módulo o Dispositivo para Programar las diferentes funciones del robot que no requiera de un computador tradicional
- Diseñar y Evaluar consumo de memoria, tiempo de CPU y consumo de energía de diferentes algoritmos implantando la Arquitectura Supsumption en los Krotic nueva versión.
- Diseñar y evaluar módulos de conexión inalámbrica para los robots considerando costo, consumo de energía, protocolo de comunicación, consumo y uso de memoria.
- Diseñar y evaluar un módulo para incorporar una cámara al Krotic, costo, consumo de energía y protocolo de comunicación.
- Diseñar una propuesta para un plan piloto en 3 escuelas para 20 niños Nacional.
- Construir 10 prototipos del Krotic nueva versión.
- Realizar 2 pruebas de campo con 10 robots

3 Marco Teórico

3.1 Antecedentes

El proyecto nace tanto del producto resultante del proyecto de investigación inscrito en la VIE Krotic: Kit de robótica Costarricense 2008-2010 [1] (Villegas-Lemus, 2010), como también de los resultados de investigaciones posteriores para mejorar las prestaciones de dicho kit de robótica.

Krotic en particular fue diseñado como un robot de bajo costo para ambientes de aprendizaje de conceptos de ciencias. El objetivo era conservarlo por debajo de \$150. Objetivo que se cumplió pues el robot terminó con un costo de \$75 dólares con funciones limitadas. Las mejoras realizadas posteriormente se obtuvieron en aspectos físicos como: apariencia y tamaño para manipulación, además en: adecuación a proyectos, autonomía energética, movilidad del robot como también y no menos para la programación. Los resultados en las experiencias de campo obtenidos con los prototipos construidos (4), los comentarios recibidos de los participantes tanto de maestras como de estudiantes durante el proceso de aplicación parcial en ambientes de aprendizaje en la modalidad después de escuela han motivado el planeamiento del presente proyecto.

3.2 Robots en el aula

El desarrollo de instrumentos como robots con fines educativos son concordantes con estrategias nacionales como la Estrategia Siglo XXI [2] (Grupo Estrategia Siglo XXI, 2006), porque no solo involucra educación, sino que apunta específicamente en educación en ciencias y además, no solo se queda en el ámbito de computación o la electrónica sino que apoya disciplinas como la pedagogía, con el enfoque constructivista como filosofía educativa propuesto por Papert [3] (Papert, 1993). Dos de los cimientos establecidos por el plan de acción de la estrategia Siglo XXI son:

- cimiento 2: la ciencia y la tecnología ligadas a la innovación y
- cimiento 4: la articulación entre lo científico, lo tecnológico y la innovación con las otras disciplinas del conocimiento social y de las humanidades y entre todas estas y la cultura y la sociedad en general.

De hecho están inherentes en la primera acción redactada en el tomo número 1 del documento Estrategia Siglo XXI.

Las propuestas tecnológicas para el uso y funciones de los robots en la educación han evolucionado desde que Papert en los años 70, cuando propuso su tortuga programable,

fundamento del lenguaje LOGO, como ocurre con LEGO Mindstorms y PARALLAX con los Boebots. Estos robots son bastante atractivos y se aplican pedagógicamente usando el construccionismo como lo hace la Fundación Omar Dengo. El principal problema es el costo. El promedio de padres de chicos de escuelas y colegios públicos no puede financiar un kit de estos para permitir a sus hijos más exploración y a los docentes de ciencias permitir el acceso para usarlos en sus clases.

Existen estudios cuyos resultados justifican el uso de robots como herramienta educativa para promover en los niños el interés por la ciencias y la tecnología, se ha estudiado el impacto del uso de robots tal como los realizados por Yousuf [4] (Yousuf, 2009) y Balch [5] (Tucker Balch, 2008) en donde se muestran las implicaciones de hardware, software y obviamente en el aspecto pedagógico también a nivel universitario.

El enfoque pedagógico adoptado por el proyecto, por los resultados obtenidos anteriormente es el construccionismo, ya que efectivamente como afirma Papert en [3] (Papert, 1993) el aprendizaje se hace pertinente a los estudiantes, también el mismo autor ha estudiado el impacto en el trabajo colaborativo y la promoción de ideas innovadoras cuando los problemas y en el entorno son pertinentes a los estudiantes.

A nivel global se reconoce que los robots son herramientas de gran potencial, que se pueden incluir en la aulas de clases de los diferentes niveles, como lo demuestran diferentes proyectos en Europa, tal como TERECoP en [7] (Dimitris Alimisis, 2006) donde se adoptó en construccionismo como fundamento pedagógico y en USA con grupos de investigación en robótica como en MIT Media Laboratory con Martin y posteriormente con Resnick con el grupo “Lifelong Kindergarden”.

Otro problema en el país, es el costo de mantenimiento, pues debe ser tal que se pueda acondicionar de las necesidades y reducidos presupuestos locales, en particular del MEP si se desea lograr el nivel de impacto nacional propuesto. Un producto que sirva de herramienta para la educación en particular de ciencias básicas como la física, que permitan el aprendizaje de diferentes tecnologías y que por su costo se pueda emplear a escala apoya los dos cimientos definidos en la estrategia.

Para mantener bajos, así como el consumo de energía y memoria de cada algoritmo propuesto, ya que estos dispositivos usan baterías y traen una memoria medida en unidades de kilobytes.

3.3 Arquitectura de Software para Robots

En cuanto las arquitecturas para el robot, el propuesto en la investigación es la “Subsumption” propuesta por Brooks en [8] (Brooks, 2000).

De acuerdo con Russell y Norvick (11) (Stuart J. Russell, 2010) las arquitecturas de software de un robot deben combinar el control reactivo con el modelo de base de la planificación deliberativa en modelos llamados de arquitectura híbrida. El control reactivo, como su nombre lo indica, está dirigido por los sensores y es apropiado para decisiones sencillas de bajo nivel en tiempo real, sin embargo, no es muy adecuado para como una solución de gobierno global del robot, puesto que esta requiere Deliberación (“deliberation”). En el contexto de robótica Deliberación se entiende como el proceso de producción de planes para una tarea (asignación) específica y a la búsqueda del plan más adecuado para hallar para realizar la tarea, sin ninguna garantía del resultado (Night R Fisher F) (12) (Russell Knight, 2001). Para realizar el trabajo de planificación deliberativa, es necesario que el robot cuente con tiempo al momento de tomar una decisión y de buenos recursos de procesamiento tales como memoria y velocidad de procesador.

Como el robot Krotic está en desarrollo, se debe lograr primero una plataforma estable de software para cumplir con los aspectos reactivos en tiempo real, experimentar para producir las especificaciones necesarias para el desarrollo de niveles más altos de la arquitectura de software del robot. Es por ello que se seleccionó la arquitectura Subsumption para el Krotic, lo que supone algoritmos reactivos simples.

Subsumption propuesto por Brooks es una estructura para relacionar máquinas de estado finito (FSM) con controladores de reacción del robot. Los estados de estas FSM pueden contener variables de sensores y las transiciones se pueden asociar con las acciones a tomar si cuando se presentan las condiciones de transición ya sean con otras FSM o con los dispositivos del robot (por ejemplo motores o luces). Sin embargo, la falta del proceso de deliberación de la arquitectura Subsumption, la dependencia en gran medida de los sensores, la dificultad de programación de la interacción de múltiples FSM, hace que la arquitectura Subsumption se circunscriba a robots con tarea simple, por ejemplo seguir una luz o seguir un trazado específico evitando obstáculos.

Dentro de las arquitecturas híbridas la más popular es la Arquitectura de Tres Capas (Three Layer Architecture). Esta arquitectura está compuesta por una capa reactiva muy en contacto con los sensores, las decisiones de este nivel están se dan en términos de milisegundos. Una capa ejecutiva o de secuencia de control de acciones, que permite directivas desde la capa deliberativa, las organiza en secuencia de acciones de la capa reactiva, las decisiones de esta capa están en términos de segundos. Por último la capa deliberativa esta genera planes para una solución global de problemas complejos, algunas veces en esta capa o nivel se incorporan modelos de toma de decisiones.

Existen otras arquitecturas que ejecutan procesamiento paralelo tales como la Pipeline, pero esas están más allá del alcance de esta investigación.

3.4 Máquinas Virtuales

Para la implantación de software tanto para el Krotic como para el Módulo o Dispositivo de Programación, se estructuró cada función como una instrucción de un conjunto de instrucciones de sendas máquinas virtuales. Esto de acuerdo con la definición planteada por Tanenbaum en [9] (Tanenbaum, 2006), ella consiste en una máquina hipotética M1 construida sobre una máquina real M0. La máquina M1 tiene asociado un lenguaje de máquina L1 y cuando se escribe un programa en ese lenguaje, éste se traduce al lenguaje de máquina de nivel inferior (L0). La traducción puede ser: interpretada, compilada o una combinación de ambos. De esta forma Tanenbaum también introduce el concepto de Máquinas Multinivel, cuando se tienen una serie de capas de máquinas con el fin de obtener funciones cada vez más complejas. El concepto de máquina Multinivel facilita el estudio y también el desarrollo de los sistemas computacionales, ya que se puede trabajar por niveles, lograr especialización para hacer eficiente un determinado nivel, obteniendo una mejora global en el desempeño del sistema si entrar a rediseñar cada uno de los niveles.

3.5 Unidad de Control Microprogramada

Para desarrollar cada uno de los módulos se va aplicar la estructura de una unidad de control microprogramada para la ejecución de instrucciones como se describe en Stallings en [10] (Stallings, 2013). De esta forma queda muy fácil agregar, modificar o eliminar funciones que no tengan buen desempeño. Se realizarán las pruebas acorde con un plan de pruebas.

Stallings define Unidades de Control (UC) en términos de los estados del procesador. La ejecución de las instrucciones se divide en varios ciclos o estados del procesador. Para el diseño de la unidad de control es necesario, tomar cada estado o ciclo del procesador y atomizarlo en operaciones sencillas llamadas microoperaciones, en la que generalmente están involucrados registros internos del procesador.

Algunos estados típicos son Ciclo de Fetch, Ciclo de Indirecto, Ciclo de Ejecución y Ciclo de Interrupción cada uno de los estados tiene asociado códigos para los ciclos de Instrucción o estado del procesador, por ejemplo:

00 Ciclo de Fetch

01 Ciclo de Indirecto

10 Ciclo de Ejecución

11 Ciclo de Interrupción

En cuanto a las Implementaciones de las Unidades de Control (UC), las más conocidas son las alambradas y la Microprogramadas. Las Unidades de Control Alambradas.

En las U.C. alambradas básicamente se toman las señales de entradas y se construye la tabla de verdad con las salidas requeridas en la secuencia requerida mediante Decoders. El reloj debe garantizar la propagación de la señal en cada una de las unidades funcionales.

Este esquema es rápido, sin embargo presenta el problema que para actualizaciones o modificaciones del set de instrucciones es necesario rediseñar completamente el sistema, pues las salidas con una función lógica de las entradas.

Unidades de Control Microprogramadas, el término de UC Microprogramada fue acuñado por M. Wilkes a principio de los años 50. Sin embargo su uso no aplicado comercialmente hasta la IBM/360 de acuerdo con Stallings [10] (Stallings, 2013). Un conjunto de microinstrucciones para implementar la ejecución de una instrucción o parte de ella se llama Microprograma. La especificación de instrucciones se hace mediante un lenguaje de microprogramación.

Esquema General de una UC:

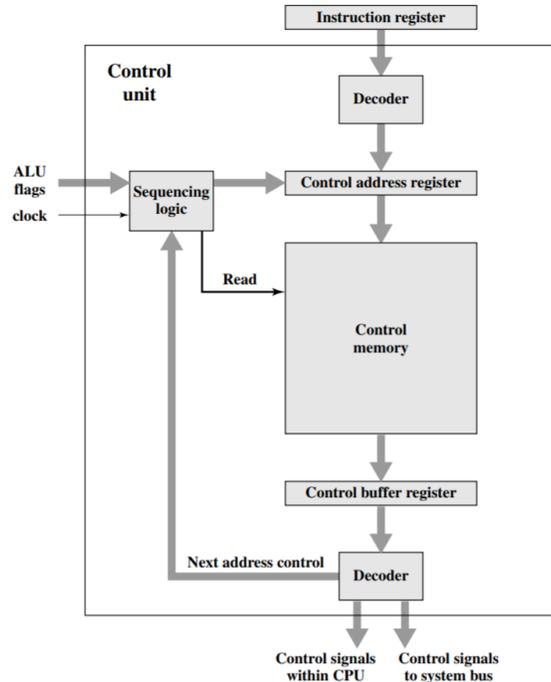


Figura 1 Funcionamiento de una Unidad de Control Microprogramada según Stallings (Stallings, 2013).

Dentro de las ventajas de la unidad de control Alambradas están: la facilidad de mantenimiento para incluir nuevas instrucciones en el conjunto de instrucciones o modificaciones a las instrucciones existentes y la otra es la facilidad de implementación de instrucciones complejas.

El proyecto propone el uso del esquema de Unidad de Control Microprogramada, para implementar un Set de instrucciones asociados a Krotic.

Otorgando un set de instrucciones a Krotic permite que esté modularizado de tal forma que se pueda adecuar a la mayor cantidad de proyectos posibles que propongan los chicos en las escuelas. Esto va a requerir una evaluación desde el punto de vista de consumo energético de algoritmos, pues con este enfoque no hay representación central del mundo donde se mueve el robot más aun no hay un sistema central que gobierne el robot, además de los recursos tradicionales de memoria y CPU.

3.6 Descripción de los principales dispositivos evaluados:

3.6.1 WiFi-Shield:

El WiFi-shield es un módulo fabricado por Arduino, dicho módulo permite la conectividad del Arduino al internet de forma inalámbrica, adicionalmente es una plataforma “open source” y es programable por medio de una biblioteca creada por Arduino destinada a controlar dicho modulo desde el microcontrolador presente en el Arduino por medio del protocolo SPI. La biblioteca creada por Arduino permite crear una conexión cliente servidor y enviar datos por medio de la red, además soporta los protocolos TCP y UDP. En la figura 1 se puede observar una imagen del dispositivo.



Figura 2 Imagen de módulo Wifi-Shield

3.6.2 Módulo RF24:

El módulo RF24 es un módulo creado por terceros, diseñado para funcionar por medio del protocolo SPI conectado a un microcontrolador como el ATMEGA 328P presente en el Arduino, el cual posee una biblioteca que permite crear un protocolo de transferencia de datos por medio de radio frecuencia. El protocolo diseñado para este módulo se basa en la existencia de un nodo “master” y los demás nodos conectados se catalogan como “slave”, los cuales envían datos al master y el máster decide lo que debe hacer con esos datos, o simplemente envían datos de un nodo al otro por medio del master que funciona como puente en la comunicación. En la figura 2 se puede observar una imagen del dispositivo.

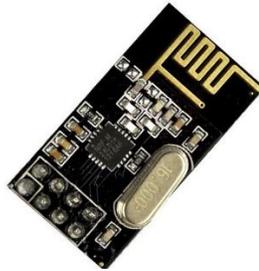


Figura 3 Imagen de módulo RF24

4 Metodología

La metodología usada para el diseño de la investigación Krotic versión Extendida, está inspirada en la Metodología Agile, que agrupa un conjunto de modelos de desarrollo incrementales e iterativos [1] (Agile Coaching, 2013), características que se adecuan para la investigación y desarrollo de sistemas empotrados. A partir de estos métodos o estructuras de trabajo (“frameworks”) se han rescatado:

- El desarrollo rápido de propuestas de hardware y software.
- La estructura no jerárquica dentro del grupo de trabajo.
- Las pruebas de las partes con los potenciales usuarios.
- La rápida adaptabilidad en el desarrollo

Es importante destacar que se han tomado algunos principios, pero no se ha adoptado todos, solo los que se consideraron prácticos para el Laboratorio LuTec y para este proyecto de investigación en particular. También se tomaron conceptos de la teoría general de los sistemas, en particular lo referente a sistemas de aro cerrado.

Un insumo importante para la especificación de este proyecto se heredó de los resultados del proyecto anterior.

Dentro de los principales criterios de diseño que se plantearon fueron el costo, consumo de energía y espacio de memoria del programa monitor y el de usuario.

El método general adaptado que se describe a continuación, se aplicó al hardware, al software y a la propuesta para el uso del Krotic:

Como todo sistema cerrado se describe en términos de entradas, proceso, salidas, retroalimentación y es naturalmente iterativo.

Entradas:

Retroalimentación de versiones previas, necesidad de usuarios en campamento.

Procesos:

Especificación

Requerimientos de las plataformas de microcontroladores tanto para el programador (tricorder) como para el cubo.

Especificación del Set de Instrucciones

Diseño

Establecer Criterios de selección de circuitos y diseño de los mismos.

Establecer esquema de la máquina virtual

Establecer el Set de Instrucciones

Configuración

Construcción de la propuesta de prototipo hardware/software

Mediciones

Validación Pruebas

Si hay elementos para corregir devolverse a especificación

Salida

Construcción del Prototipo Final

En la figura 4 se puede apreciar el diagrama de flujo del método empleado en esta investigación para los módulos principales y cada uno de los submódulos del proyecto.

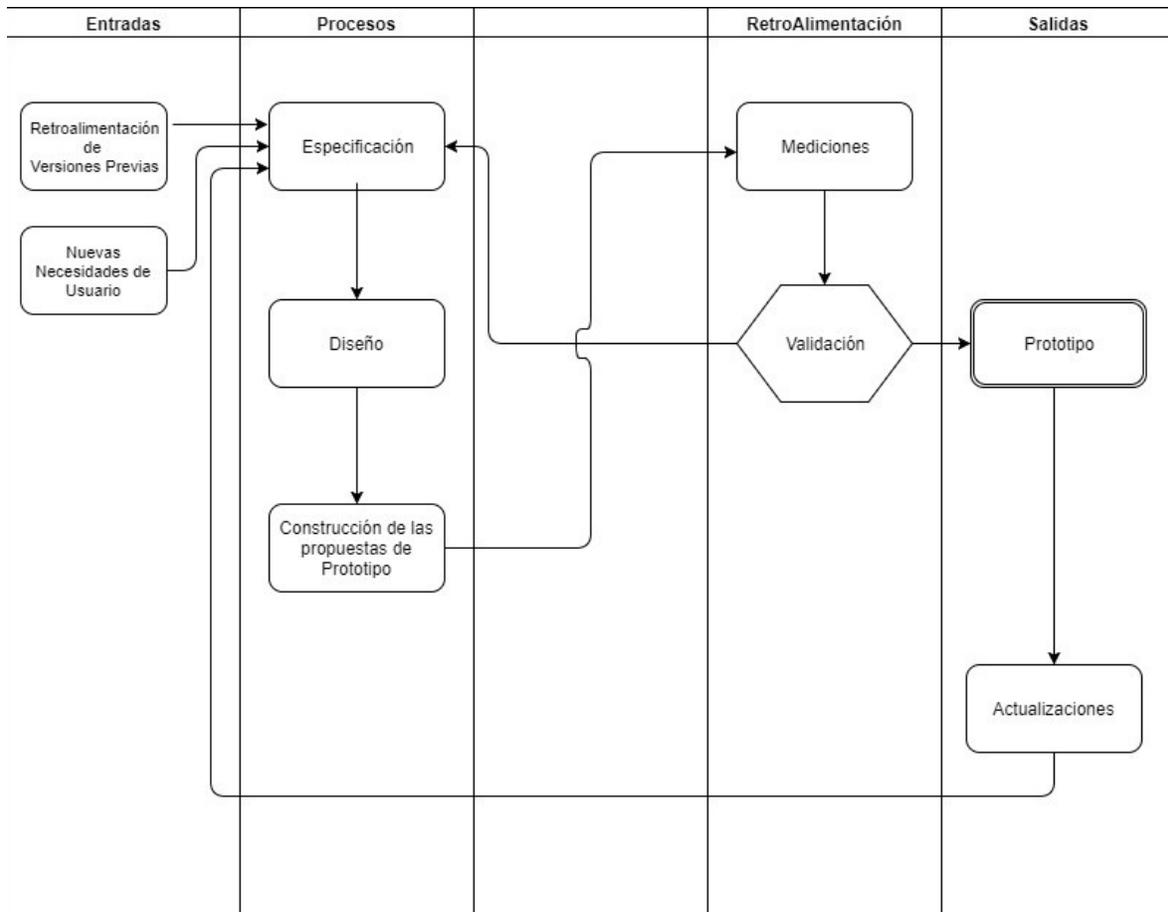


Figura 4 Adaptación Metodología Agile aplicada al proyecto.

4.1 Software

Con base en la arquitectura de Subsumption se establecieron una las funciones propias para el kit de robótica. Se especificaron los algoritmos a evaluar y se especificaron diferentes pruebas para cada uno de los casos. Para la implantación de software tanto para el Krotic como para el Módulo o Dispositivo de Programación se estructuró cada función como una instrucción de un conjunto de instrucciones de una máquina virtual para cada caso. Para desarrollar cada uno de los módulos se aplicó la estructura de una unidad de control microprogramada para la ejecución de instrucciones. De esta forma queda muy fácil agregar, modificar o eliminar funciones que no tengan buen desempeño. Se realizaron las pruebas de programación y transferencias acorde con un plan de pruebas.

4.2 Hardware

Se establecieron los diseños de circuitos de acuerdo con la especificación de las funciones de software. Se diseñaron los esquemáticos, posteriormente se implantaron los prototipos, se realizaron las diversas pruebas y mediciones. Se realizarán en forma iterativa los ajustes necesarios y después se pasaron a la construcción de los prototipos

En cuanto a la Integración Hardware/Software para cada prototipo de hardware montado se ejecutó el software correspondiente ya depurado en la pruebas de software. Se realizaron las mediciones y se pasó a la etapa de evaluación. Se ensamblaron construyeron los prototipo con los resultados de mejor desempeño.

En lo correspondiente al Diseño del Factor de Forma, de acuerdo con las dimensiones de los componentes se pasó a diseñar las estructuras para el Krotic, el dispositivo de programación y los módulos complementarios del kit. Se construyeron dos prototipos de cada uno para de los dispositivos principales, estos son: los módulos de cámara e inalámbrico y el Módulo o Dispositivo de Programación. Con dichos módulos se hizo la evaluación y pruebas de ajuste de los componentes e integración del hardware software, se trabajó también en la distribución física de los componentes y los conectores.

Se construyeron los 10 prototipos con base en los resultados de la evaluación y experimentación anterior. Se definió la logística para la realización de 2 pruebas parciales previas a las pruebas con los 10 robots. Se estableció la logística para la realización de las 2 pruebas de campo definidas para el proyecto. Ello se realizó en dos de las escuelas que han sido visitadas en los últimos 4 años.

El proyecto planteó usar una arquitectura Subsunción para desarrollar 2 módulos físicos nuevos, uno de ellos con cámara para de esta forma establecer un mecanismo de reconocimiento adicional de su ambiente y el otro un módulo de comunicación WiFi. Se requirió del diseño desarrollo y construcción de dispositivos para programar los robots sin usar un computador convencional y la construcción de prototipos de los robots completos un plan piloto para mediciones. Estas tecnologías comprometen el consumo de energía y la autonomía del robot, además de que aumentan el costo, el cual se busca mantener lo más bajo posibles.

5 Resultados

Se relacionan los resultados de los principales módulos del proyecto de investigación:

Programador, Krotic, Cámara y Módulo Inalámbrico.

5.1 Programador

Como producto de la realización de pruebas de campo con los módulos desarrollados en la generación anterior, se llegó a la conclusión de que era plausible que construyendo un módulo programable tipo menú como modelo mental, éste se podría adaptar a las diferentes necesidades de los proyectos planteados por los chicos, haciéndolos así más versátiles. Además se detectó que era importante que la propuesta del programador no requiriera un computador en su presentación usual, es decir tipo escritorio, portátil o de mano, para que los chicos pudieran configurar los módulos.

El programador permite entonces portabilidad, fácil replicación contando con los planos y programas. Es decir una institución, facilitador, padre o interesado podría tomarlos y reproducirlos a bajo costo.

Para la implementación dichos módulos se seleccionaron microcontroladores ya que son procesadores de bajo costo y consumo de energía y facilita la capacidad de programación.

Por sus consecuencias en tiempo de implantación, curva de aprendizaje y desempeño; la elección del microcontrolador también debe realizarse tomando en cuenta factores de consumo y costo.

Al finalizar la etapa de evaluación se seleccionó un microcontrolador Atmel de la serie Tiny, más específicamente el 13. Con respecto al lenguaje o plataforma de desarrollo a usar, surgió un desafío en cuanto al desarrollo del software del proyecto, en especial cuando se limita el consumo de energía y en general cuando se impone un límite de costo bajo. Por lo anterior, considerando las limitaciones de consumo, las necesidades de optimización de memoria, su manejo en el programa y la velocidad o tiempo de respuesta, todo esto prácticamente determinó que el lenguaje seleccionado para programar fuera ensamblador Tanenbaum [8] (Tanenbaum, 2006) en lugar de opciones como un compilador de C.

Para la estructura del software, se decidió conceptualizar la programación de cada módulo, como la ejecución de una instrucción que pertenece a un conjunto de instrucciones

de una máquina virtual según Tanenbaum [9] (Tanenbaum, 2006). Al hacerlo de esta forma, queda muy fácil conceptualizar el arreglo de rutinas y conectar el flujo de ejecución con el concepto de unidades de control microprogramadas, en una descripción como la mostrada en Stallings [10] (Stallings, 2013). Para ejecutar el conjunto de instrucciones tienen que hacer uso de una serie de operaciones llamadas microinstrucciones que están almacenadas en porciones específicas de memoria llamada micro ROM y además llevar un apuntador de la secuencia de operación denominado por algunos autores como Stallings [12] (Stallings, 2013) y fabricantes como micro PC, llamado micro contador de programa.

Cómo plan de pruebas se estableció que el comportamiento funcional de cada módulo debía obedecer mínimo el comportamiento de los módulos ya existentes en el laboratorio LuTec en consumo, nivel de señal, opciones de configuración de entradas y salidas, construcción y programación. Para ello se aplicaron para cada módulo 3 pruebas de montaje y diseño por parte de chicos y chicas, la cantidad fue variable para cada módulo.

Los resultados de las pruebas de campo fueron muy positivos y plantean nuevos desafíos para incorporar nuevas funciones cuyo equivalente actual serían nuevos módulos. El programador y sus módulos mostraron ser realmente versátiles al incorporarlos en los proyectos planteados por los chicos. El manejo del programador mismo pareció ser estimulante para la población a la que se le aplicó ya que mostraron interés por aprender su manipulación. Se debe fortalecer el aspecto mecánico de la conexión de los módulos, ya que este fue un punto débil que resaltó durante las pruebas.

A continuación, en la tabla 1, se presenta el SET de Instrucciones de Krotic, versión 2014 diseñado, desarrollado e implementado para el robot. Igualmente se construyó el compilador para este SET de instrucciones.

Tabla 1 Set de Instrucciones del Krotic

Set de Instrucciones del Krotic

Ver 1.2 al 2/12/2014

Descripción de registros:

- Tamaño de registro: 16 bits.
- Se tienen 32 registros disponibles: Desde D00 a D1F.
- LSB a la derecha y MSB a la izquierda.

Instrucciones de Movimiento:

Código	Mnemónico	Modo direccionamiento	Descripción
01	MOT_FOR DISTANCE	Inmediato	Avanza por distancia
02	MOT_FOR TIME	Inmediato	Avanza por tiempo
03	MOT_FOR100D	Inherente	Avanza 100 unidades de distancia
04	MOT_FOR1S	Inherente	Avanza por 1 segundo
05	MOT_BAC DISTANCE	Inmediato	Retrocede por distancia
06	MOT_BAC TIME	Inmediato	Retrocede por tiempo
07	MOT_BAC100D	Inherente	Retrocede 100 unidades de distancia
08	MOT_BAC1S	Inherente	Retrocede por 1 segundos
09	MOT_ROT CW ANGLE	Inmediato	Rota ANGLE grados en sentido horario
0A	MOT_ROT CW TIME	Inmediato	Rotar en sentido horario por TIME segundos.
0B	MOT_ROT CW 90D	Inherente	Rota 90 grados en sentido horario
0C	MOT_ROT CW 180D	Inherente	Rota 90 grados en sentido horario
0D	MOT_ROT CW 1S	Inherente	Rota por un segundo en sentido horario
0E	MOT_ROT CCW ANGLE	Inmediato	Rota ciertos grados en sentido anti horario

0F	MOT_ROTCCW TIME	Inmediato	Rotar en sentido anti horario por determinado tiempo
10	MOT_ROTCCW90D	Inherente	Rota 90 grados en sentido anti horario
11	MOT_ROTCCW180D	Inherente	Rota 90 grados en sentido anti horario
12	MOT_ROTCCW1S	Inherente	Rota por un segundo en sentido anti horario
13	MOT_FOR_R DN	Inmediato	Avanza por distancia
14	MOT_FOR_R DN	Inmediato	Avanza por tiempo
15	MOT_BAC_R DN	Inmediato	Retrocede por distancia
16	MOT_BAC_R DN	Inmediato	Retrocede por tiempo
17	MOT_ROTCCW_R DN	Inmediato	Rota ciertos grados en sentido horario
18	MOT_ROTCCW_R DN	Inmediato	Rotar en sentido horario por determinado tiempo
19	MOT_ROTCCW_R DN	Inmediato	Rota ciertos grados en sentido anti horario
1A	MOT_ROTCCW_R DN	Inmediato	Rotar en sentido anti horario por determinado tiempo

Instrucciones de Input Sensores (Un solo pin)

20	INPUT_BIN PIN, DN	Registro inmediato	DN←Lectura de sensor base 2 (binaria)
21	INPUT_QUA DN	Registro inmediato	DN←Lectura de sensor base 4
22	INPUT_OCT DN	Registro inmediato	DN←Lectura de sensor base 8 (octal)
23	INPUT_HEX DN	Registro inmediato	DN←Lectura de sensor base 16 (hexadecimal)
24	INPUT_1K DN	Registro inmediato	DN←Lectura de sensor base 1K

*DN donde N es de 0 a 5 son los registros de conversión analógico digital

Instrucciones de Input Sensores (Múltiples pines)

25	INPUT_BIN_M DN, PIN0	Registro inmediato	DN←Lectura de sensor base 2 (binaria)
26	INPUT_QUA_M DN, PIN0, PIN1	Registro inmediato	DN←Lectura de sensor base 4
27	INPUT_OCT_M DN, PIN0, PIN1, PIN2	Registro inmediato	DN←Lectura de sensor base 8 (octal)
28	INPUT_HEX_M DN, PIN0, PIN1, PIN2, PIN3	Registro inmediato	DN←Lectura de sensor base 16 (hexadecimal)
29	INPUT_1K_M DN, PIN0,...,PIN7	Registro inmediato	DN←Lectura de sensor base 1K

*RN donde N es de 0 a 5 son los registros de conversión analógica digital

Instrucciones de Input por carga

30	LOAD DN, VALOR		DN←Se carga el valor
31	LOAD_IC DN		DN←Se carga el valor que viene del registro paralelo 74LS166

*DN donde N de 0 a 15 son los posibles registros de datos disponibles al usuario (variables)

Instrucciones de Output

3A	OUT PIN0, DN		Saca el valor de un registro disponible al usuario a PIN0 del micro.
3B	OUT DN, PIN0, PIN1		Saca el valor de un registro disponible al usuario a PIN0 Y PIN1 del micro.
3C	OUT DN, PIN0, PIN1, PIN2, PIN3		Saca el valor de un registro disponible al usuario a PIN0, PIN1, PIN2 y PIN3 del micro.
3D	OUT DN, PIN0,...,PIN7		Saca el valor de un registro disponible al usuario a PIN0,..., PIN7 del micro.
3E	OUT_IC DN		Saca un valor de un registro DN de datos a un 74LS164

Nota: Se debe tener correspondencia con respecto a los pines ya ocupados por el sistema.

Instrucciones de Transferencia

40	MOV DN0,DN1	Registro Inmediato	DN0←DN1
----	-------------	--------------------	---------

Instrucciones de Aritméticas

41	INC DN		DN←DN+1
42	DEC DN		DN←DN-1
43	ADD DN,VALOR		DN←DN+VALOR
44	SUBS DN,VALOR		DN←DN-VALOR
45	ADD DN0,DN1		DN0←DN0+DN1
46	SUBS DN0,DN1		DN0←DN0-DN1

Instrucciones Condicionales

4A	IFCOND DN, VALOR		DN-VALOR si 0 PC+1 Si diferente PC+4
4B	IFCOND DN0, DN1		DN0-DN1 si 0 PC+1 Si diferente PC+4
4C	IFDELAY SEGUNDOS		Trascurridos segundos Ejecuta PC+1
4D	IFDELAY DN		Trascurridos segundos Ejecuta PC+1

Instrucciones de Acciones

50	HALT		KILL ALL
51	SEND DN	Registro inmediato	Envía un registro por WiFi
53	PWM DN	Registro inmediato	Simula salida analógica

Rutina de prueba:

- 1) Motor activado con sensor de luz:

```
INPUT_BIN DN
IFCOND DN, 1
MOT _FOR TIME
IFDELAY SEGUNDOS
```

- 2) Encender un leds con sensor de luz:

```
INPUT_BIN D0
LOAD D1, 000F
LOAD D2, 0000
IFCOND D0, 1
OUT D1, PIN0, PIN1, PIN2, PIN3
IFDELAY 0003
OUT D2, PIN0, PIN1, PIN2, PIN3
IFDELAY 0003
```

5.2 Pruebas de duración y consumo durante quemado de Sketch:

Se realizaron pruebas de consumo y tiempo de duración del proceso de grabación en EEPROM (“quemado”) de un conjunto de instrucciones (predefinidas) de alto nivel en el microcontrolador ATmega-328P; para dicho procedimiento se utilizó un circuito de grabación de memoria E²PROM (“quemador”), para seleccionar los códigos de instrucción y los argumentos. Además se utilizó una plataforma RaspberryPi descrita anteriormente, en el mismo se ejecuta el programa quemador del microcontrolador. Adicionalmente se alimentaron los circuitos con dos baterías de la misma marca pero diferente capacidad y se realizaron dos iteraciones para medir consumo de los circuitos conectados y los tiempos de duración de quemado de la rutina que se va a describir posteriormente. Se describe a continuación las condiciones generales para la presente fase de pruebas:

- Lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica.
- Toma de Mediciones: Asistente Byron Rojas.
- Descripción del lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica en el Laboratorio LuTEC, ubicado en segundo piso del edificio.

- Descripción del ambiente: Ambiente seco y climatizado a 20 grados Celsius.
- Baterías utilizadas: dos baterías:
 - A) Batería Digital Energy (9.6V/1600mAh)
 - B) Batería Digital Energy (9.6V/2000mAh)
- Multímetros utilizados:
 - A) Multímetro 1 (T1): Meterman 10xL
 - B) Multímetro 2 (T2): Pro's Kit 303150NCS
- Rutina utilizada:

Tabla 2 Códigos de programa de prueba

Instrucción	Argumento	Descripción
44	Pin 7	Enciende un pin digital
42	1 segundo	Retrasa la ejecución n segundos
45	Pin 7	Apaga un pin digital
42	1 segundo	Retrasa la ejecución n segundos

5.2.1 Primera Iteración:

Se cargaron las baterías durante tres horas, el tiempo recomendado para su óptima utilización, posteriormente se procedió a conectar el circuito programador y el RaspberryPi a la batería, luego se corrió el programa quemador y se procedió a medir las tensiones de la batería con los circuitos conectados a la misma, posteriormente una vez desconectados, además se utilizó un cronómetro para medir la duración en minutos del proceso de quemado de la rutina durante cierto número de iteraciones, se finalizó el proceso de lectura en el momento que el RaspberryPi, se comenzó a reiniciar por falta de energía y no pudo continuar ejecutando con el proceso de quemado de la rutina.

A continuación se muestran las tablas que resumen los datos observados durante las pruebas, donde se muestra el número de iteración que corresponde al número de veces que se quemó el sketch, luego los valores mínimos y máximos observados durante la práctica, la tensión de la batería medida con el circuito conectado y el circuito desconectado; finalmente se observa una columna con los datos del tiempo que se duró desde el inicio del proceso de quemado hasta que el Raspberry Pi dejó de funcionar correctamente. (La marca y modelo de las baterías se especifican anteriormente).

En la tabla 3, valores obtenidos para la Batería A:

Tabla 3 Mediciones Batería A, iteración 1

Iteración	Corriente Min (mAh)	Corriente Max (mAh)	Tensión T1 (V)	Tensión T2 (V)	Tensión con carga (V)	Duración (min:seg)
1	660	740	10,05	10,01	9,93	0:05
10	650	720	9,97	9,92	9,92	1:10
20	640	730	9,89	9,86	9,84	2:11
30	570	680	9,81	9,77	9,75	3:24
40	560	670	9,74	9,69	9,64	6:09
50	580	710	9,64	9,6	9,5	6:14
100	550	780	9,53	9,55	9,32	7:03
150	-	-	7,23	7,3	9,25	8:15

En la tabla 4, valores obtenidos para la Batería B:

Tabla 4 Mediciones Batería B, iteración 1

Número	Corriente Min (mAh)	Corriente Max (mAh)	Tensión T1 (v)	Tensión T2 (v)	Tensión Durante (v)	Duración (min)
1	740	770	10,16	10,07	9,85	0:03
10	720	760	10,14	10,04	9,78	1:05
20	570	770	10,05	10,1	9,74	2:10
30	610	720	10	10,06	9,3	3:30
40	740	740	9,9	9,94	9,24	6:05
50	740	760	9,87	9,91	9,18	9:10
100	740	780	9,76	9,81	9,02	6:27
150	740	780	9,51	9,48	8,85	8:32
150	740	770	9,05	8,99	6,93	8:30
200	-	-	7,03	7,08	4,93	0:00

5.2.2 Segunda Iteración:

Se cargaron las baterías nuevamente durante tres horas, posteriormente se procedió nuevamente a conectar el circuito programador y el RaspberryPi a la batería, luego se corrió el programa quemador para corroborar los datos de la primera iteración con los datos de la segunda iteración, se midió corriente y tensión de la batería del circuito conectado y el circuito sin conectar, luego se cronometró nuevamente el tiempo de duración de quemado de diferente número de rutinas en el microcontrolador y finalmente se detuvo la medición en el momento que se observó el reinicio del RaspberryPi que corre el programa de quemado.

En la tabla 5, valores obtenidos para la Batería A:

Tabla 5 Medición Batería A, iteración 2

Número	Corriente Min (mAh)	Corriente Max (mAh)	Tensión T1 (V)	Tensión T2 (V)	Tensión Batería (V)	Duración
1	520	800	10,2	10,35	9,6	0:04
10	510	780	10,19	10,31	9,6	0:30
20	520	770	10,19	10,31	9,51	1:50
30	810	850	10,25	10,21	9,6	1:50
40	810	830	10,15	10,14	9,5	2:10
50	680	850	10,08	10,12	9,26	2:36
100	710	750	9,99	10,05	9,58	5:00
150	690	740	9,81	9,98	9,15	5:05
150	670	720	9,91	9,92	8,03	7:28
200	670	720	9,61	9,58	7,61	7:30
250	-	-	9,56	9,4	5,38	10:30

En la tabla 6, valores obtenidos para la Batería B:

Tabla 6 Medición Batería B, iteración 2

Número	Corriente Min (mAh)	Corriente Max (mAh)	Tensión T1 (V)	Tensión T2 (V)	Tensión Batería (V)	Duración (s)
1	670	710	10,65	10,68	10,29	0:03
10	670	700	10,5	10,52	10,3	0:30
20	680	690	10,35	10,36	10,13	0:45
30	700	750	10,29	10,31	10,02	1:30
40	700	750	10,23	10,26	9,98	2:39
50	720	770	10,17	10,17	9,9	3:15
100	610	650	10,05	10,13	9,85	6:28
150	580	630	9,99	10	9,78	9:55
150	580	630	9,85	9,85	9,59	9:44
200	710	760	9,53	9,58	9,24	12:45
250	-	-	8,87	8,87	6,21	14:48

Las lecturas sugieren que la Batería B Batería Digital Energy (9.6V/2000mAh), ofrece suficiente autonomía para sostener un tiempo de quemado más prolongado, se puede apreciar en la figura con la línea marcada.

Cuando la tensión llega a un nivel de 5.33V o inferior, no se garantiza el proceso de quemado del programa. Hay que tener en cuenta que en este proceso la carga fue continua es decir que el micro que grababa (proceso de “quemado”) continuamente durante el tiempo mostrado en las tablas. Es concordante con el valor nominal de 2A. Sin embargo, los tiempos no fueron concluyentes pues las conexiones presentaron problemas lo que pudo incidir en los tiempos para lograr quemar la cantidad de veces requerida.

Se decidió también por una diferencia de precio de 4180 colones escoger la Batería B por la relación costo vs beneficio.

En la figura 5 el diagrama del diseño final del Quemador incluyendo el RaspberryPi con un solo despliegue siete segmentos.

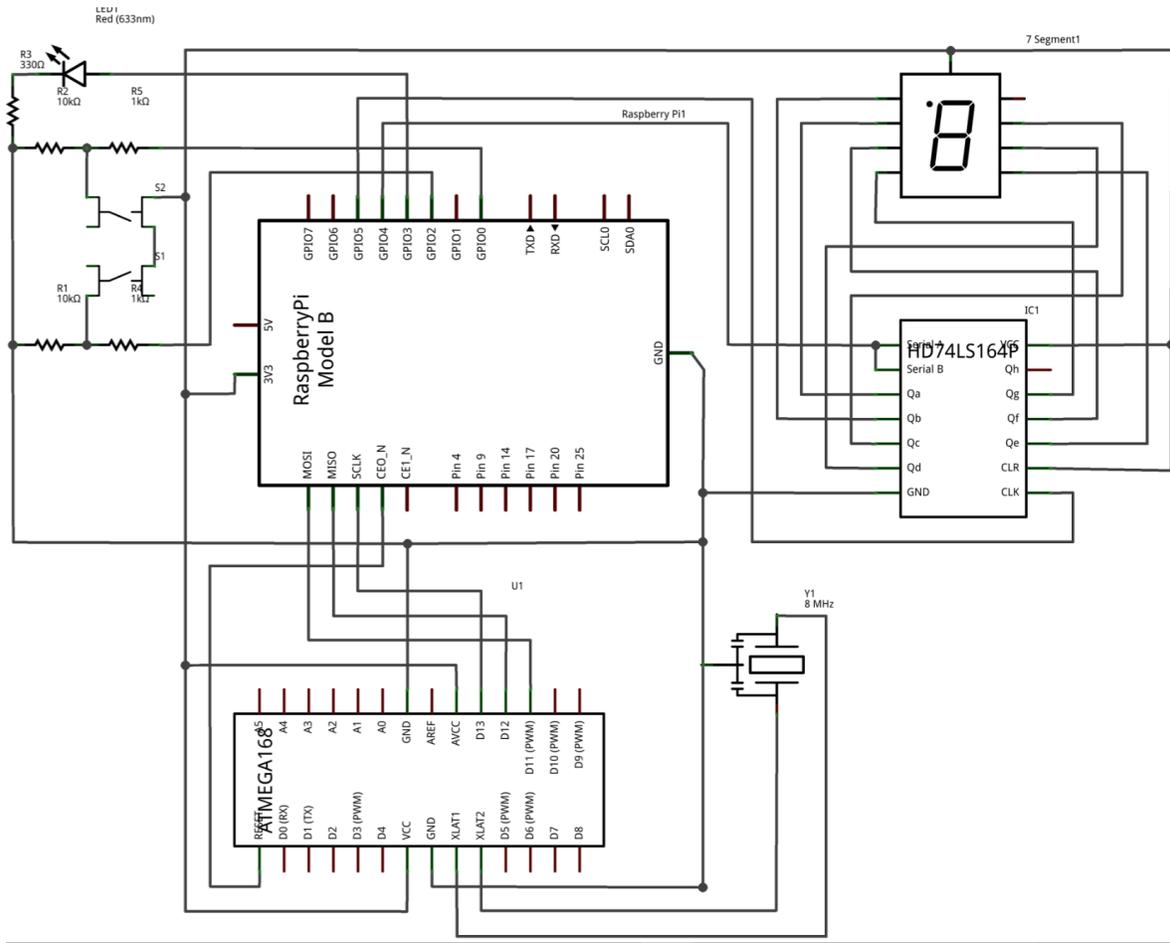


Figura 5 Diagrama esquemático del primer diseño del quemador incluyendo el RaspberryPi

El quemador construido con el diagrama anterior, no era muy efectivo en términos de usabilidad, pues cuando se realizaron las primeras validaciones con usuarios, se encontró que con un solo despliegue siete segmentos se requerían más pasos para realizar la entrada de una instrucción específica, con esta retroalimentación en la siguiente iteración se rediseñó el quemador o Tricorder con 2 despliegues, se puede apreciar en la figura 7. En la figura 6, se muestra el diagrama del último diseño del quemador con el que se condujeron las pruebas de programación.

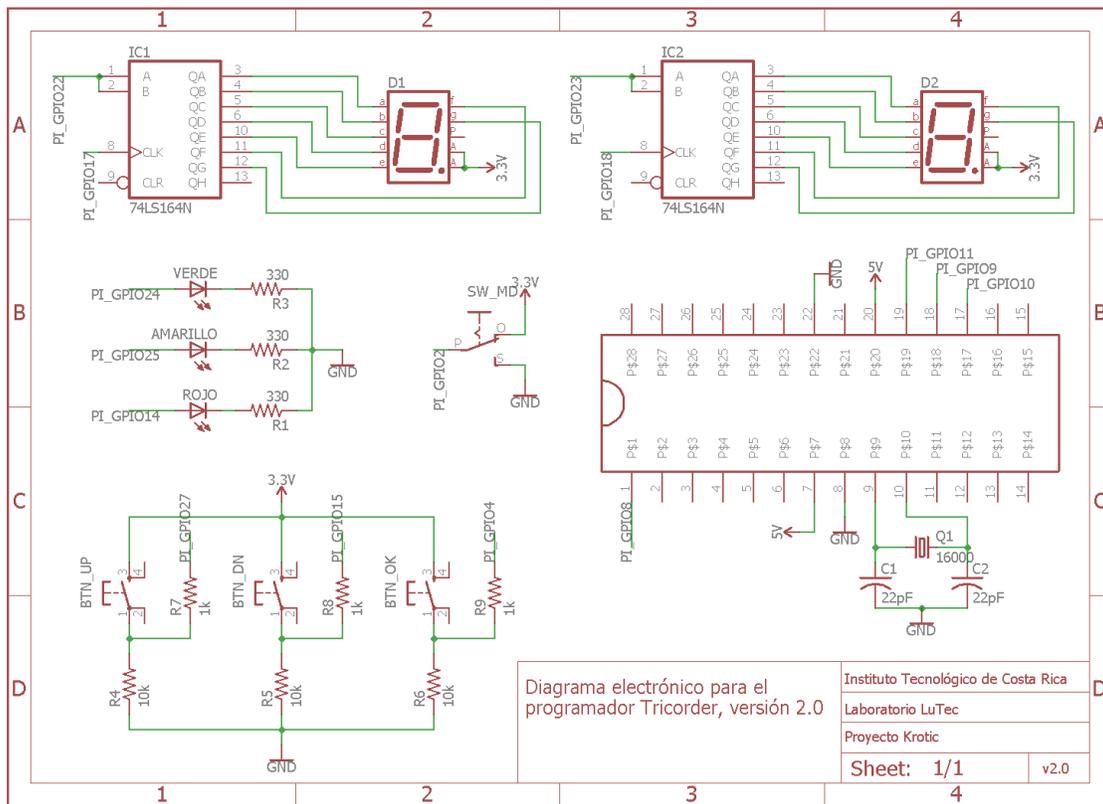


Figura 6 Diagrama electrónico esquemático del quemador con el que se construyeron los prototipos

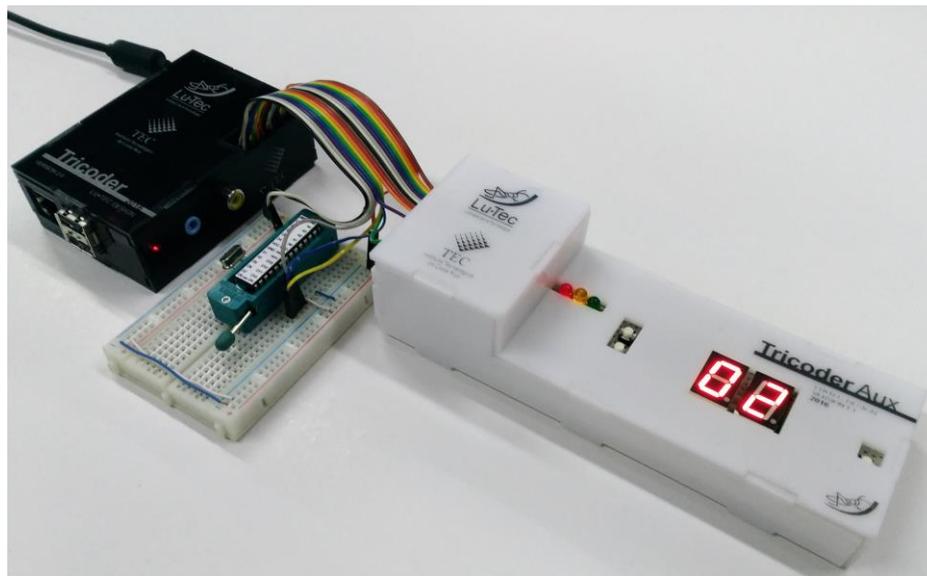


Figura 7 Tricorder conectado para transferir programa al microcontrolador del Robot

5.3 Robot Krotic

Los circuitos componentes de Krotic (Cubo), se montaron y probaron para asegurarse que las medidas de tensión, corriente y por lo tanto potencia estuviesen dentro de los parámetros aceptables de acuerdo con la especificación del Krotic anterior versión, esto es funcionamiento entre 4.5 V y 6V de tensión y un consumo de alrededor de 65 mA. Posteriormente se pasó al ensamblaje de los diferentes robots, primero 5 para ejecutar las pruebas de acuerdo con el plan. Estas consistían en hacerlos funcionar por 30 segundos para asegurar la locomoción del robot. Posteriormente se ensamblaron los otros 5 robots y se les sometió al mismo conjunto de pruebas.

El siguiente paso en las pruebas consistió en validar el lenguaje con el set de instrucciones, verificar consistencia de los diferentes programas. Es importante recalcar que el set de instrucciones del robot obedeció a la especificación realizada en la descripción funcional.

La forma de probar fue ejecutar una a una las instrucciones, si alguna no se comportaba dentro de acuerdo con la descripción se devolvía al proceso de programación para realizar los ajustes en forma iterativa hasta que cumpliera con la descripción de las instrucciones.

Una vez con el lenguaje del robot estuvo depurado se procedió a la integración del software con el hardware montando el microcontrolador con el intérprete del lenguaje del robot ya grabado en él.

Se programaron varios algoritmos del tipo reactivo de subsunción para realizar las pruebas finales. A continuación se relacionan los diferentes resultados de dichas pruebas.

Una de las variables observadas en las pruebas anteriores estaba relacionada con la fuente de alimentación. Por ello se relacionan las medidas con varios tipos de baterías.

Cabe mencionar que al final se determinó que la tensión presentaba problemas para el buen funcionamiento del robot pues hacía que el robot se reiniciara varias veces en un minuto por lo que se tomó la decisión de incrementar la tensión a 9 V. El elemento regulador de tensión LM78050 que aunque la hoja de datos no lo especifica la comprobación experimental demostró que al llegar la tensión a nivel inferior a 5V se comporta de forma ambigua y no regulara de forma estable, esto provocaba que el microcontrolador identificara un nuevo arranque y reiniciara sus rutinas.

Las pruebas de consumo se realizaron entonces con 9V.

Resumen de Pruebas de Consumo con alimentación con baterías tamaño AA convencionales y recargables.

- Lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica.
 - Toma de Mediciones: Asistente Andrés Brais.
 - Descripción del lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica en el Laboratorio LuTEC, ubicado en segundo piso del edificio.
 - Descripción del ambiente: Ambiente seco y climatizado a 21 grados Celsius.
 - Baterías utilizadas: dos baterías:
- C) Batería Digital Energy (9.6V/1600mAh)
D) Batería Digital Energy (9.6V/2000mAh)

- Multímetros utilizados:
La descripción se presenta en las tablas 8 y 9.

C) Multímetro 1

Usado para las primeras mediciones de Tensión

Tabla 7 Multímetro 1 usado en las mediciones de tensión

Descripción del multímetro	
Marca	Meterman
Modelo	10XL
Medición de tensión	20 V
Medición máxima de tensión	1000 V

D) Multímetro 2

Usado para las segundas mediciones de Tensión

Tabla 8 Multímetro 2 usado en las mediciones de tensión

Descripción del multímetro	
Marca	Pro'sKit
Modelo	303-10NCS
Medición de tensión	20 V
Medición máxima de tensión	600 V

- Rutina utilizada:

Las mediciones se realizaron en dos rondas cada uno para corroborar experimentalmente el comportamiento de acuerdo con las especificaciones.

Primera ronda de mediciones de tensión en el Krotic, se presenta en la tabla 9, con el algoritmo Subsumption programado para seguir la Luz con los sensores con batería de 9V recargable:

Tabla 9 Mediciones con batería recargable 9V, primera ronda

Batería recargable Tamaño 9V (valor nominal de 8,4V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	7,98	4,79	7,31	4,89
2	8,58	4,94	7,32	4,81
3	8,91	4,85	7,83	4,54
4	8,3	4,86	7,18	4,73
5	9,16	4,9	8,02	4,55
6	7,84	4,71	6,95	4,64
7	8,45	4,9	7,21	4,66
8	8,39	4,94	7,14	4,82
9	8,37	4,93	7,16	4,74
10	8,42	4,93	7,21	4,8

Segunda ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con batería de 9V recargable, se muestra en la tabla 10:

Tabla 10 Mediciones con batería recargable 9V, segunda ronda

Batería recargable Tamaño 9V (valor nominal de 8,4V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	8,67	4,93	8,01	4,88
2	8,65	4,91	7,02	4,71
3	8,33	4,98	7	4,84
4	7,9	4,74	7,22	4,46
5	8,43	4,87	7,13	4,51
6	7,693	4,9	7,08	4,67
7	7,85	4,75	6,94	4,68
8	8,29	4,93	6,98	4,73
9	8,33	4,9	6,63	4,6
10	8,33	4,92	6,35	4,09

En las tabla 11 se presenta la primera ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con batería de 9V convencional:

Tabla 11 Mediciones con batería convencional de 9V, primera ronda

Batería convencional de 9V (valor nominal 9V)				
Krotic en reposo		Krotic moviendose		
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	8,5	4,92	8,01	4,9
2	8,95	4,94	8,14	4,85
3	9,07	4,89	8,27	4,67
4	8,82	4,92	8,01	4,84
5	9,17	4,89	8,28	4,54
6	7,38	4,65	6,21	4,52
7	8,01	4,9	6,3	4,59
8	8,08	4,94	6,3	4,61
9	7,97	4,93	6,4	4,71
10	7,98	4,93	6,31	4,73

Segunda ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con batería de 9V convencional, los valores se presentan en la tabla 12:

Tabla 12 Mediciones con batería convencional 9V, segunda ronda

Batería convencional de 9V (valor nominal 9V)				
Krotic en reposo		Krotic moviendose		
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	8,56	4,93	7,27	4,83
2	8,8	4,92	7,97	4,78
3	8,67	5,2	7,81	5,04
4	8,3	4,8	7,69	4,46
5	8,03	4,93	6,43	<u>4,72</u>
6	7,35	4,64	6,13	4,51
7	7,23	4,58	5,97	4,31
8	7,86	4,92	7,06	4,34
9	8,03	4,92	7,01	4,28
10	8,89	4,92	6,41	4,61

Primera ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con baterías AA recargables, en la tabla 13 se presentan los valores:

Tabla 13 Mediciones batería recargable AA, primera ronda

Baterías recargables AA (valor nominal 1,2V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	7,61	4,82	7,28	4,74
2	8,05	4,86	7,04	4,41
3	7,75	4,86	6,85	4,43
4	7,84	4,8	6,42	4,58
5	7,92	4,86	7,16	4,56
6	7,98	4,78	7,61	4,67
7	8,19	4,87	7,32	4,48
8	8,14	4,91	7,5	4,63
9	8,15	4,91	7,52	4,68
10	8,08	4,91	7,46	4,69

En la tabla 14 se muestra la segunda ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con baterías AA recargables:

Tabla 14 Mediciones batería recargable AA, segunda ronda

Baterías recargables AA (valor nominal 1,2V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	7,94	4,91	7,39	4,76
2	7,87	4,85	6,63	4,44
3	7,6	4,75	5,94	3,67
4	7,9	4,73	6,97	4,43
5	8,2	4,88	7,23	4,48
6	7,73	4,79	7,41	4,68
7	7,73	4,72	7,11	4,55
8	7,95	4,87	7,26	4,57
9	7,78	4,86	6,9	4,27
10	8,1	4,89	7,44	4,65

Primera ronda de mediciones de tensión en el Krotic, se muestran los valores en la tabla 15, con el algoritmo Subsumption programado para seguir la Luz con los sensores con baterías AA convencionales:

Tabla 15 Mediciones batería convencional AA, primera ronda

Baterías convencionales AA (valor nominal 1,5V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	8,84	4,84	8,4	4,75
2	9,05	4,87	7,94	4,53
3	9,01	4,88	7,73	4,38
4	8,75	4,81	7,63	4,56
5	9,21	4,86	8,15	4,47
6	8,54	4,81	8,09	4,68
7	8,89	4,86	8,04	4,5
8	8,89	4,9	8,03	4,63
9	8,85	4,89	7,84	4,58
10	8,9	4,9	8,11	4,75

Segunda ronda de mediciones de tensión en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores con baterías AA convencionales, los datos se muestran en la tabla 16:

Tabla 16 Mediciones batería convencional AA, segunda ronda

Baterías convencionales AA (valor nominal 1,5V)				
Krotic en reposo			Krotic moviendose	
# de krotic	Tensión de la batería (V)	Tensión de alimentación (V)	Tensión de la batería (V)	Tensión de alimentación (V)
1	9,16	4,94	8,29	4,88
2	8,86	4,85	7,26	4,46
3	8,73	4,82	6,92	4,26
4	8,62	4,8	7,58	4,46
5	8,88	4,89	7,88	4,52
6	8,58	4,79	8,12	4,66
7	8,5	4,73	7,98	4,45
8	8,84	4,87	8,07	4,5
9	8,64	4,87	7,31	4,52
10	8,8	4,9	7,91	4,57

A continuación se muestran los gráficos 1 y 2, que destacan el comportamiento de interés de las variables en estudio tabuladas anteriormente, éstos son los comparativos de Tensión para un mismo tamaño con los dos tipos de fuentes de alimentación primarias y recargables dentro del Krotic. Se tomaron dos lecturas por robot utilizando diferentes instrumentos de medición.

Gráfico 1 Comparación de Fuentes de Alimentación, 9V

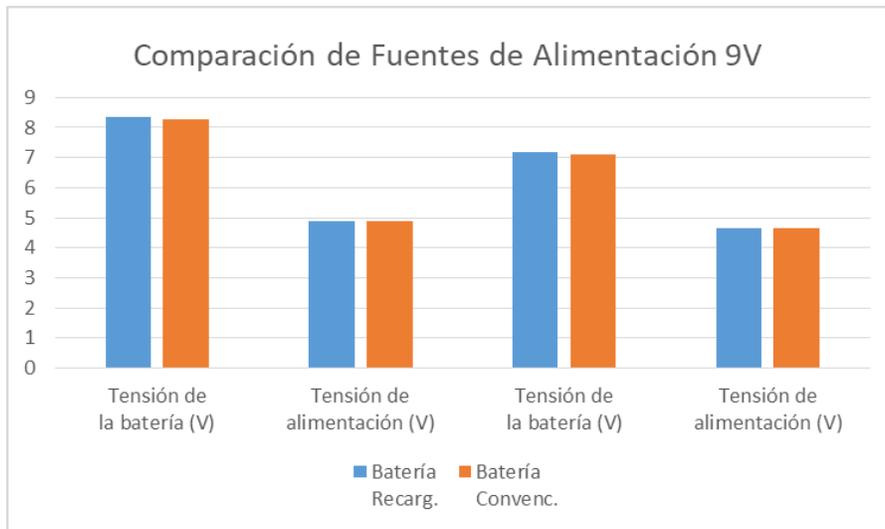
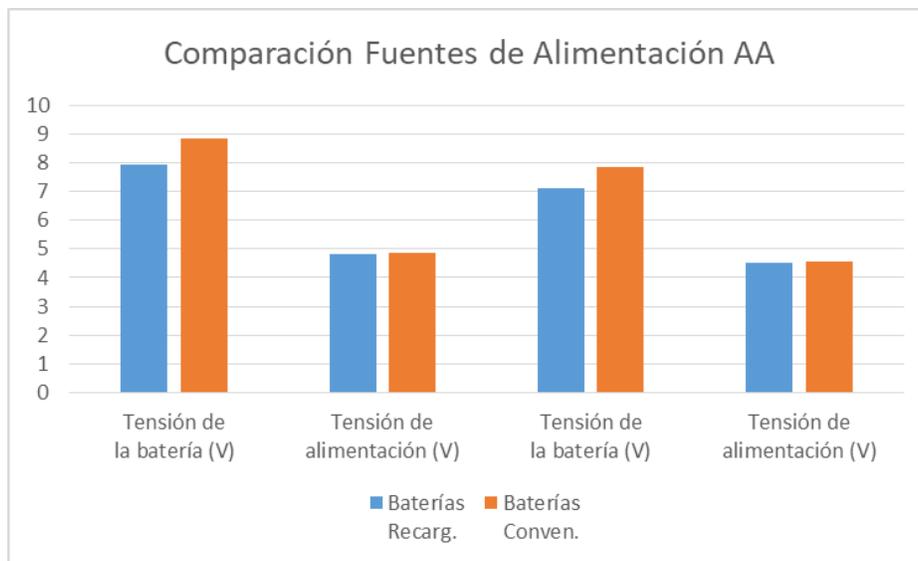


Gráfico 2 Comparación Fuentes de Alimentación, AA 8,4V



En la tabla 17, se muestra la primera ronda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías AA recargables:

Tabla 17 Mediciones de consumo en el Krotic batería recargable AA, algoritmo Subsumption seguir luz, primera ronda

Baterías recargables AA (valor nominal 1,2V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,62	34	157,08	3,56	156,3	556,428
2	4,72	35,7	168,504	4,24	184,9	783,976
3	4,92	39,9	196,308	4,41	198,4	874,944
4	4,71	53,8	253,398	4,25	182,38	775,115
5	4,98	37,9	188,742	4,46	182,6	814,396
6	4,7	102,5	481,75	4,92	210	1033,2
7	4,58	81,7	374,186	4,32	186,7	806,544
8	4,97	33,4	165,998	4,56	199,3	908,808
9	4,91	38,2	187,562	4,05	169,4	686,07
10	4,93	36,4	179,452	4,69	196,7	922,523

Segunda ronda de mediciones de Consumo en el Krotic, se muestra en la tabla 18, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías AA recargables:

Tabla 18 Mediciones de consumo en el Krotic batería recargable AA, algoritmo Subsumption seguir luz, segunda ronda

Baterías recargables AA (valor nominal 1,2V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,59	34	156,06	3,13	152,9	478,577
2	4,87	37,3	181,651	4,18	180,1	752,818
3	4,92	40	196,8	4,48	198,6	889,728
4	4,69	53,2	249,508	4,24	186,7	791,608
5	5	37,4	187	4,41	183,6	809,676
6	4,74	100,9	478,266	4,88	200	976
7	4,58	83,1	380,598	4,25	181,4	770,95
8	4,96	32,5	161,2	4,54	195,1	885,754
9	7,92	35,3	279,576	4,39	175,5	770,445
10	4,93	33,9	167,127	4,74	198,8	942,312

Primera ronda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías AA convencionales, se muestra en la tabla 19:

Tabla 19 Mediciones de consumo en el Krotic batería convencional AA, algoritmo Subsumption seguir luz, primera ronda

Baterías convencionales AA (valor nominal 1,5V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,67	35,8	167,186	3,64	159,6	580,944
2	4,86	37	179,82	4,42	190,8	843,336
3	4,9	35,2	172,48	4,97	210	1043,7
4	4,73	66,6	315,018	4,53	196,3	889,239
5	4,96	38,3	189,968	4,72	197,2	930,784
6	4,79	101,5	486,185	4,89	200	978
7	4,6	87,8	403,88	4,64	197,3	915,472
8	4,98	34	169,32	4,72	193,6	913,792
9	37,2	4,92	183,024	4,77	188,5	899,145
10	4,93	33,6	165,648	4,76	197,9	942,004

Segunda ronda de mediciones de Consumo en el Krotic, se muestra en la tabla 20, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías AA convencionales:

Tabla 20 Mediciones de consumo en el Krotic batería convencional AA, algoritmo Subsumption seguir luz, segunda ronda

Baterías convencionales AA (valor nominal 1,5V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,64	34	157,76	3,43	156,4	536,452
2	4,87	38	185,06	4,33	184,5	798,885
3	4,95	38,8	192,06	4,87	210	1022,7
4	4,76	67,4	320,824	4,54	196,5	892,11
5	4,97	38,5	191,345	4,7	193,5	909,45
6	4,8	106,6	511,68	210	4,89	1026,9
7	4,61	90,4	416,744	4,64	198,1	919,184
8	4,99	33,1	165,169	4,73	196,2	928,026
9	4,9	35,8	175,42	4,75	185,7	882,075
10	4,93	33,3	164,169	4,76	195,6	931,056

En la tabla 21 se muestra la primera ronda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías 9V recargables:

Tabla 21 Mediciones de consumo en el Krotic batería recargable 9V, algoritmo Subsumption seguir luz, primera ronda

Batería recargable Tamaño 9V (valor nominal de 8,4V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,6	35,4	162,84	3,67	158,6	582,062
2	4,81	39,7	190,957	4,15	179,8	746,17
3	4,94	38,3	189,202	4,85	200	970
4	4,79	66,9	320,451	4,58	1976,6	9052,828
5	4,95	38	188,1	4,66	195,9	912,894
6	4,83	99,2	479,136	4,69	210	984,9
7	4,61	89,2	411,212	4,64	198,5	921,04
8	4,96	32,7	162,192	4,69	196,2	920,178
9	4,9	36,8	180,32	4,63	183	847,29
10	4,93	35,1	173,043	4,74	197,6	936,624

Segunda ronda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías 9V recargables, se muestra en la tabla 22:

Tabla 22 Mediciones de consumo en el Krotic batería recargable 9V, algoritmo Subsumption seguir luz, segunda ronda

Batería recargable Tamaño 9V (valor nominal de 8,4V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,62	34,2	158,004	3,64	162,8	592,592
2	4,83	36,1	174,363	4,44	188,3	836,052
3	4,94	39,4	194,636	4,87	200	974
4	4,76	67,3	320,348	4,63	199,4	923,222
5	4,95	37,7	186,615	4,66	192,6	897,516
6	4,81	97,8	470,418	4,88	200	976
7	4,62	88,7	409,794	4,63	199,1	921,833
8	4,95	33	163,35	4,64	198,9	922,896
9	4,91	36,3	178,233	4,65	183,6	853,74
10	4,93	34,6	170,578	4,75	195,7	929,575

Primera ronda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías 9V convencionales, se muestra en la tabla 23:

Tabla 23 Mediciones de consumo en el Krotic batería convencional 9V, algoritmo Subsumption seguir luz, primera ronda

Batería convencional de 9V (valor nominal 9V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,7	33,8	158,86	3,6	158,3	569,88
2	4,84	36,5	176,66	4,17	168,7	703,479
3	4,94	38,5	190,19	4,22	192,2	811,084
4	4,7	58,5	274,95	3,77	163,5	616,395
5	4,94	38	187,72	3,65	154,7	564,655
6	4,49	73,8	331,362	188,6	4,22	795,892
7	4,44	91,8	407,592	3,96	172,1	681,516
8	4,92	34,6	170,232	3,95	176,5	697,175
9	4,9	35,9	175,91	4,27	176,6	754,082
10	4,92	34,5	169,74	4,32	187,4	809,568

Segunda tanda de mediciones de Consumo en el Krotic, con el algoritmo Subsumption programado para seguir la Luz con los sensores, con baterías 9V convencionales, se muestra en la tabla 24:

Tabla 24 Mediciones de consumo en el Krotic batería convencional 9V, algoritmo Subsumption seguir luz, segunda ronda

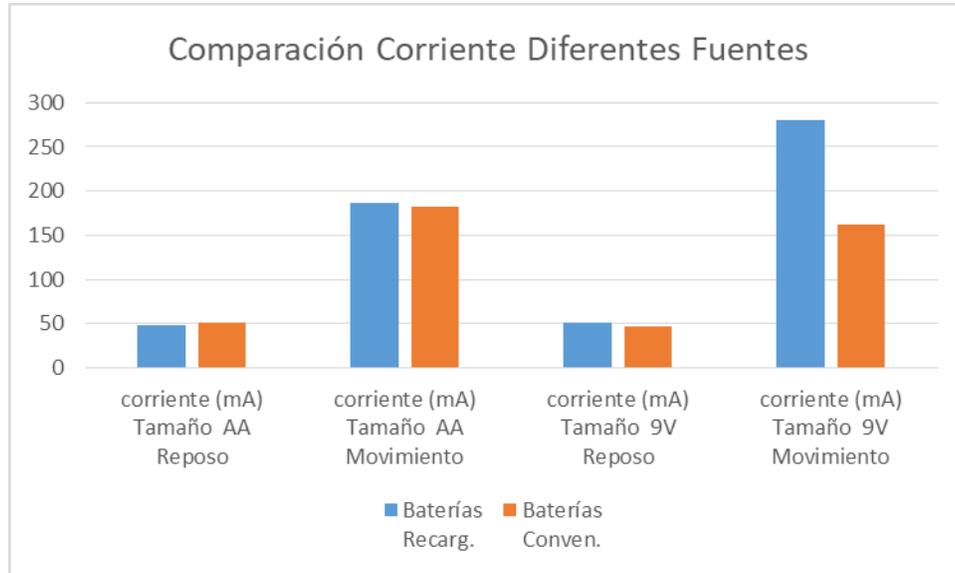
Batería convencional de 9V (valor nominal 9V)						
Número de cubo	Krotic en reposo			Krotic en movimiento		
	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)	Medición de tensión (V)	Medición de corriente (mA)	(Consumo) Potencia (mW)
1	4,6	33,8	155,48	3,55	154,1	547,055
2	4,79	36,4	174,356	4,14	167,8	694,692
3	4,94	38,1	188,214	4,08	181,3	739,704
4	4,55	66,6	303,03	4,19	176,3	738,697
5	4,94	37,8	186,732	3,56	150,3	535,068
6	4,76	65,4	311,304	4,16	187,2	778,752
7	4,4	84,2	370,48	3,69	164	605,16
8	4,91	34,6	169,886	4,04	179,8	726,392
9	4,9	37,6	184,24	4,13	171,4	707,882
10	4,92	34,4	169,248	3,7	158,3	585,71

Prueba de potencia de los servomotores de los robots que muestran consumo atípico

Comparación de la corriente en el Krotic dependiendo para los dos tamaños estudiados AA y 9V en diferentes situaciones en reposo o ejecutando un algoritmo tipo Subsumption para seguir una luz.

El gráfico 3 muestra que las baterías recargables comparadas, entregan más corriente que las convencionales.

Gráfico 3 Comparación Corriente Diferentes Fuentes



Comparación del consumo cuando se ejecuta el algoritmo seguir una luz empleando dos sensores. Las mediciones se realizaron con los dos tipos de baterías para los dos tamaños seleccionados para el Krotic.

El gráfico 4 muestra que las baterías AA presentan un mejor perfil de consumo para adoptarlas en el robot Krotic.

Gráfico 4 Comparación Consumo Diferentes Corrientes

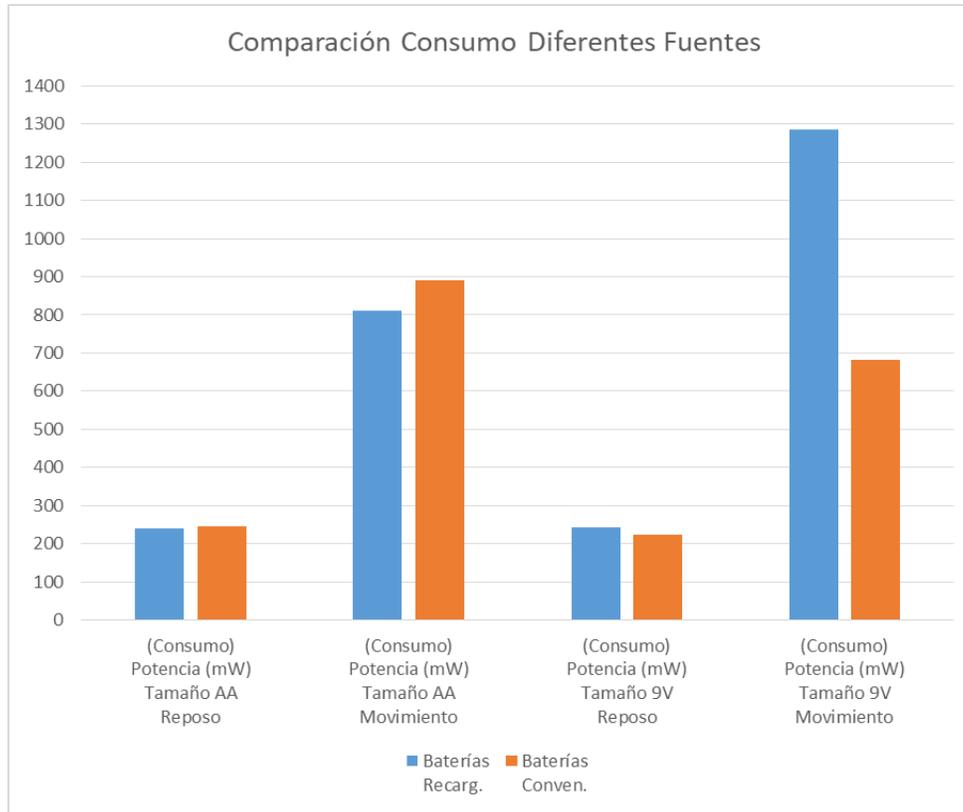


Tabla 25 Comprobación de consumo máximo del servo comercial

Número de Cubo	servo 1 (mA)		servo 2 (mA)		Tensión (V)	Estado del krotic
	estático	movim.	estático	movim.		
1	9,7	80	9,6	79,7	4,92	Cumple con la potencia
4	35,1	86,1	9,6	84,5	4,9	No cumple con la potencia
7	9,6	90,8	36,1	82,7	4,85	No cumple con la potencia
6	40,4	90,1	42,8	95,5	4,73	No cumple con la potencia
9	9,8	80,9	9,7	86,2	4,9	Cumple con la potencia

La tabla 25 revela que los servos asociados a los robots número 4, 7 y 6 presentan un consumo de potencia anómalo llegando en algunos casos a cuadruplicar el valor.

La figura 8 presenta el resultado de la construcción de los prototipos del robot y su dispositivo programador. Note el zócalo verde que es de tipo ZIF (cero fuerza inserción) para recibir el microcontrolador del robot.

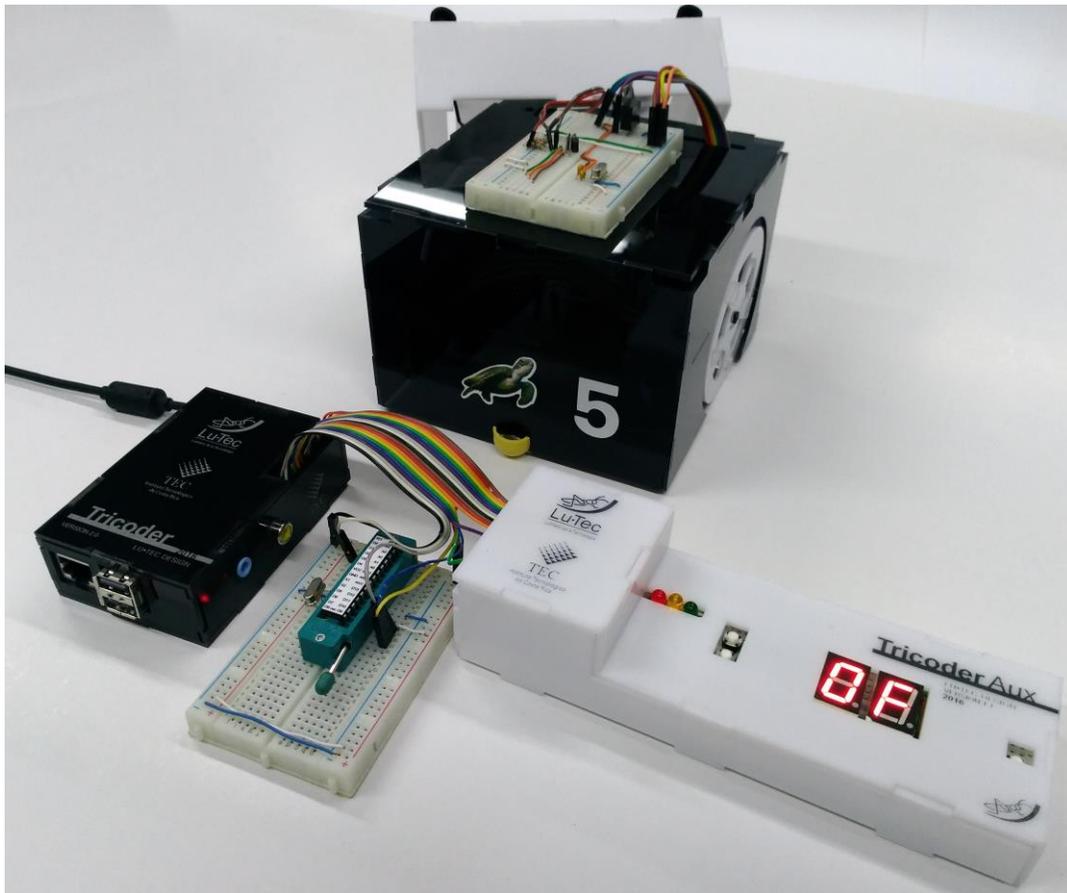


Figura 8 Robot Krotic en el proceso de grabación de E2PROM

Ejemplos de los algoritmos Subsumption programados con el lenguaje del robot versión del set de instrucciones 2.1. En virtud del conjunto de instrucciones, el diagramas de estado finitos para cada algoritmo, es bastante simple, por lo que únicamente se presenta el código correspondiente al conjunto de instrucciones diseñado para Krotic.

Algoritmo #1 – Seguidor de luz

```
MOT_FOR_UNTIL 0, 128  
MOT_CW_UNTIL 1, 128  
MOT_CCW_UNTIL 2, 128
```

Algoritmo #2 – Evita obstáculo con aprendizaje

```
IFCOND_EQ_R 5, 0, 3  
INPUT_1K 1  
INPUT_1K 2  
INC 5  
IFCOND_GT_R 1, 2, 1  
INC 3  
IFCOND_LT_R 1, 2, 1  
INC 4  
MOT_FORD 10  
IFCOND_GT_R 3, 4, 5  
MOT_ROT CW90D 0  
MOT_FORD 11  
MOT_ROT CCW90D 0  
MOT_FORD 20  
IFDELAY 5  
IFCOND_LT_R 3, 4, 5  
MOT_ROT CCW90D 0  
MOT_FORD 11  
MOT_ROT CW90D 0  
MOT_FORD 20  
IFDELAY 5  
INPUT_1K 1  
INPUT_1K 2  
ADDR 3, 1  
ADDR 4, 2  
IFDELAY 5
```

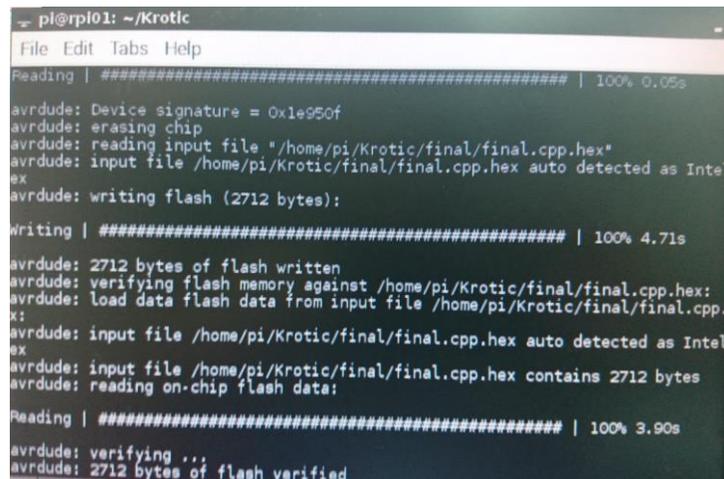
Algoritmo #3 – Evita obstáculos sin aprendizaje

```
MOT_FORD 10
```

MOT_ROTCCW90D 0
MOT_FORD 11
MOT_ROTWCW90D 0
MOT_FORD 30
IFDELAY 15

Tamaño de los programas cuando se graban desde el tricorder

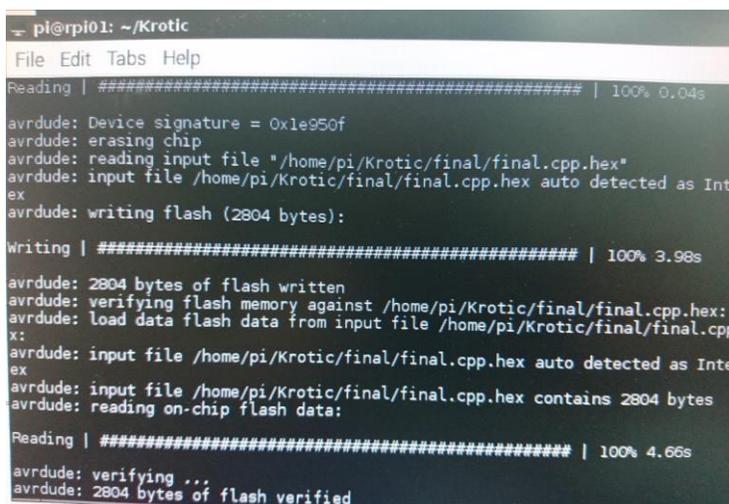
El tamaño del programa de seguimiento de luz es de 2712 bytes. En la figura 9 se muestra el tamaño del programa al quemar un Krotic.



```
pi@pi01: ~/Krotic
File Edit Tabs Help
Reading | ##### | 100% 0.05s
avrdude: Device signature = 0x1e950f
avrdude: erasing chip
avrdude: reading input file "/home/pi/Krotic/final/final.cpp.hex"
avrdude: input file /home/pi/Krotic/final/final.cpp.hex auto detected as Intel
ex
avrdude: writing flash (2712 bytes):
Writing | ##### | 100% 4.71s
avrdude: 2712 bytes of flash written
avrdude: verifying flash memory against /home/pi/Krotic/final/final.cpp.hex:
avrdude: load data flash data from input file /home/pi/Krotic/final/final.cpp.f
x:
avrdude: input file /home/pi/Krotic/final/final.cpp.hex auto detected as Intel
ex
avrdude: input file /home/pi/Krotic/final/final.cpp.hex contains 2712 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 3.90s
avrdude: verifying ...
avrdude: 2712 bytes of flash verified
```

Figura 9 Programa de seguimiento de luz

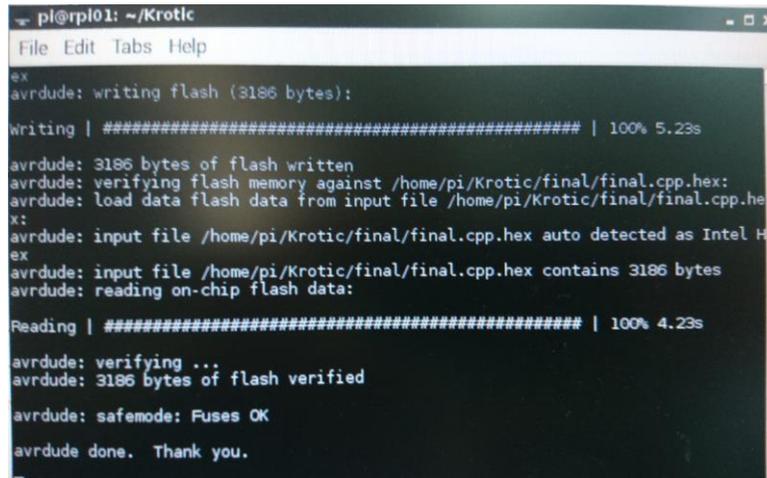
El tamaño del programa de tomar un camino sin decisión es de 2804 bytes. En la figura 10, se muestra el tamaño al quemarlo en el microcontrolador del Krotic.



```
pi@pi01: ~/Krotic
File Edit Tabs Help
Reading | ##### | 100% 0.04s
avrdude: Device signature = 0x1e950f
avrdude: erasing chip
avrdude: reading input file "/home/pi/Krotic/final/final.cpp.hex"
avrdude: input file /home/pi/Krotic/final/final.cpp.hex auto detected as Inte
ex
avrdude: writing flash (2804 bytes):
Writing | ##### | 100% 3.98s
avrdude: 2804 bytes of flash written
avrdude: verifying flash memory against /home/pi/Krotic/final/final.cpp.hex:
avrdude: load data flash data from input file /home/pi/Krotic/final/final.cpp
x:
avrdude: input file /home/pi/Krotic/final/final.cpp.hex auto detected as Intel
ex
avrdude: input file /home/pi/Krotic/final/final.cpp.hex contains 2804 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 4.66s
avrdude: verifying ...
avrdude: 2804 bytes of flash verified
```

Figura 10 Programa sin decisión

El tamaño del programa de tomar un camino con decisión es de 3186 bytes. En la figura 11 se muestra el tamaño al quemarlo en robot Krotic.



```
pi@rpi01: ~/Krotic
File Edit Tabs Help
ex
avrdude: writing flash (3186 bytes):
writing | ##### | 100% 5.23s
avrdude: 3186 bytes of flash written
avrdude: verifying flash memory against /home/pi/Krotic/final/final.cpp.hex:
avrdude: load data flash data from input file /home/pi/Krotic/final/final.cpp.he
X:
avrdude: input file /home/pi/Krotic/final/final.cpp.hex auto detected as Intel H
ex
avrdude: input file /home/pi/Krotic/final/final.cpp.hex contains 3186 bytes
avrdude: reading on-chip flash data:
Reading | ##### | 100% 4.23s
avrdude: verifying ...
avrdude: 3186 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done. Thank you.
```

Figura 11 Programa con decisión

5.4 Módulo de Comunicación Inalámbrica

Para el proyecto (Morales, 2014) realizó en LuTec un trabajo de desarrollo e implementación de una red inalámbrica para el proyecto de investigación Krotic versión extendida. Producto de este trabajo se seleccionaron dos posibles opciones de conexión para el Módulo inalámbrico Krotic siguiente versión Módulo WiFi-Shield y Módulo de Nordic RF24. De todas las pruebas realizadas se relacionan a continuación únicamente la comparación de diferentes patrones de bytes transmitidos para las dos opciones en dos distancias diferentes: 2 metros y 11.85 metros. También se incluyó en la prueba de transmisión una tecnología emergente al momento la de PCduino.

Pruebas de autonomía utilizando el Módulo WiFi-Shield y Módulo RF24:

En esta sección se resumen y comparan las observaciones realizadas durante un proceso de transferencia de patrones de bytes, entre los dos dispositivos conectados mediante un protocolo establecido específicamente para cada plataforma. En primera instancia se utilizó la plataforma ARDUINO UNO y un protocolo de comunicación Cliente-Servidor entre dos de estos dispositivos utilizando el módulo WiFi Shield (para habilitar la conexión WiFi), además se utilizó la misma plataforma ARDUINO UNO, empleándolo como medio de comunicación, obviamente dos módulos RF24 (Rx , Tx) y su protocolo de comunicación mediante una biblioteca proporcionada por el fabricante. En ambos casos, se realizó la transferencia de tres patrones, que se detallarán más adelante y se utilizaron dos distancias de referencia para realizar las mediciones.

A continuación se explican las consideraciones generales para esta prueba en específico:

- Lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica.
- Descripción del lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica en el Laboratorio LuTEC, ubicado en segundo piso del edificio.
- Descripción del ambiente: Ambiente seco y climatizado a 20 grados Celsius.
- Baterías utilizadas:
 - A) Batería Digital Energy (9.6V/1600mAh)
 - B) Batería Digital Energy (9.6V/2000mAh)
 - C) Baterías recargables 1.2v 1200 mAh.
- Descripción de patrones:
 - 1) Patrón 1: 1111111000000000 (mitad 1 y mitad 0)
 - 2) Patrón 2: 1010101010101010 (unos y ceros alternados)
 - 3) Patrón 3: 1111000011110000 (cuatro unos y cuatro ceros)

Nota: Es un ejemplo específico de 16 bits, se debe adaptar para diferentes tamaños de cadena.

- Distancias de medida:
 - 1) Distancia 1: 2m
 - 2) Distancia 2: 11.85m

Pruebas de transferencia de patrones variando tamaño y distancia desde el punto de acceso:

Pruebas WiFi-Shield:

- Distancia 1, los valores obtenidos se presentan en la tabla 26:

Tabla 26 Mediciones utilizando WiFi-Shield a 2m del servidor

Patrón	Tiempo (ms)				
	32B	64B	128B	256B	512B
1	1015	1906	2921	4954	9016
2	1012	1015	2158	4189	9142
3	1895	2023	3166	4562	9169

- Distancia 2, los valores se aprecian en la tabla 27:

Tabla 27 Mediciones utilizando WiFi-Shield a 11.85m del servidor

Patrón	Tiempo (ms)				
	32B	64B	128B	256B	512B
1	1779	2796	4830	9027	15642
2	1010	2028	4062	6097	10292
3	894	1016	2033	4957	9025

Pruebas utilizando módulo RF24:

- Distancia 1, valores se muestran en la tabla 28:

Tabla 28 Mediciones utilizando módulo RF24 a 2m del servidor

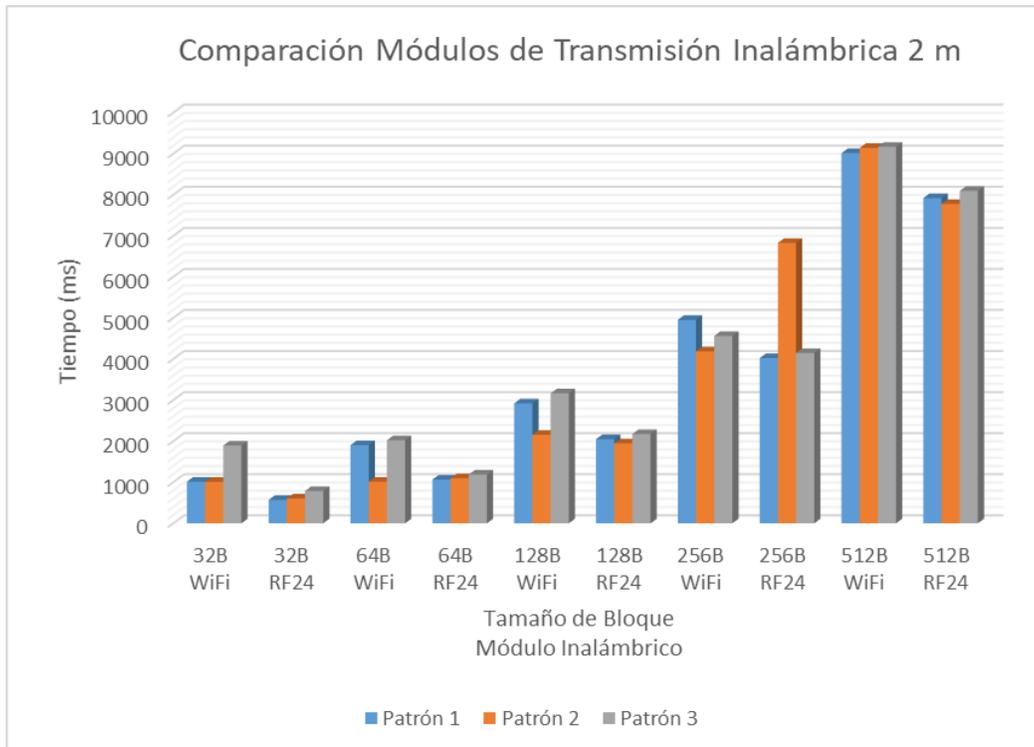
Patrón	Tiempo (ms)				
	32B	64B	128B	256B	512B
1	570	1064	2051	4025	7922
2	606	1099	1947	6828	7776
3	788	1188	2175	4148	8096

- Distancia 2, observar tabla 29:

Tabla 29 Mediciones utilizando módulo RF24 a 11.85m del servidor

Patrón	Tiempo (ms)				
	32B	64B	128B	256B	512B
1	652	1145	1951	6635	11055
2	462	955	1943	3916	7744
3	605	1098	2085	4059	7913

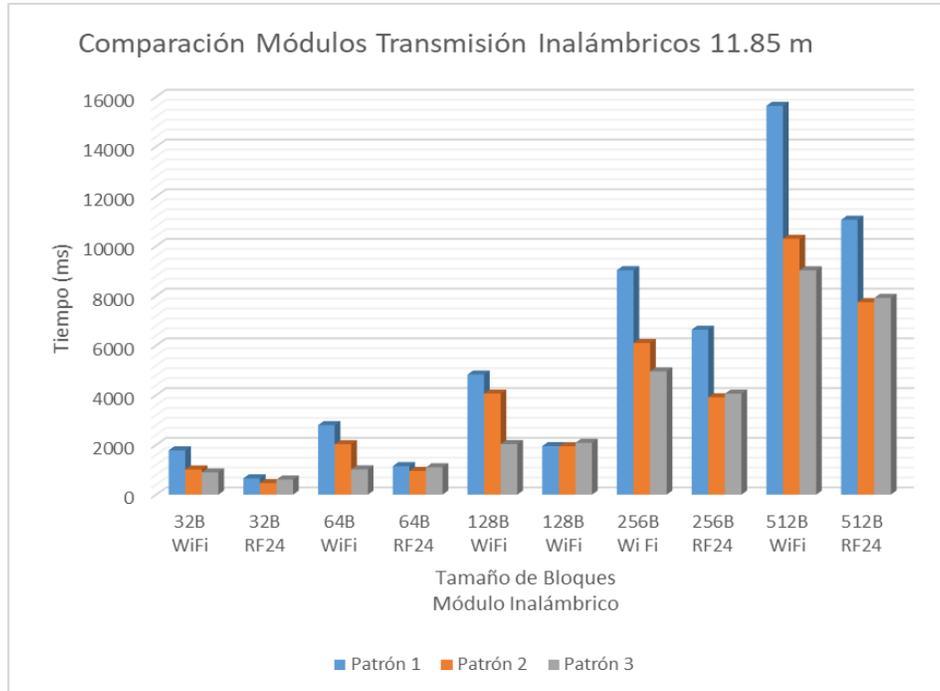
Gráfico 5 Comparación Módulos de Transmisión Inalámbrica 2m



El Gráfico 5, muestra un comparativo entre los Módulos WiFi Arduino y RF24 con diferentes patrones de bytes a una distancia de 2 metros.

El Gráfico 6, resume una comparación entre los Módulos WiFi Arduino y RF24 con diferentes patrones de bytes a una distancia de 11.85 metros.

Gráfico 6 Comparación Módulos Transmisión Inalámbricos 11.85 cm



Pruebas realizadas utilizando las plataformas Raspberry pi y PCduino, además variando la distancia desde punto de acceso:

Inicialmente, se procedió a implementar un set de pruebas, que permitiera comparar el rendimiento entre las dos plataformas y además una aproximación de la máxima capacidad de cada una de ellas, analizando un conjunto significativo de parámetros que facilitara definir una elección clara de plataformas, para un propósito específico.

En este caso se decidió implementar un protocolo de red inalámbrica WIFI, que permite establecer una conexión eficiente entre dispositivos de diferentes plataformas y además se requirió que cada dispositivo tuviese la capacidad de enviar y recibir comandos o instrucciones de forma genérica por medio de la red.

Fue necesario realizar mediciones que consideraran tres parámetros: Consumo de energía (típico en este tipo de sistemas), tasas de transferencia y detección de errores. Para la implementación de este protocolo en específico, se establecieron dos plataformas a utilizar como servidor, además, una plataforma a utilizar como cliente. Se utilizaron las

plataformas “Raspberry PI” y “PcDuino v2” como servidores y un Arduino UNO con su respectivo “WIFI SHIELD” como cliente. Además, se añadió un microcontrolador de Arduino ATMEGA 328P sin su tarjeta para ayudar en el analizar su comportamiento.

Del lado del servidor se utilizó el lenguaje Python 3.2.2, así como sus respectivas bibliotecas para medición de tiempos y conexión de ‘sockets’ de red. Al lado del cliente se utilizó el lenguaje C de alto nivel utilizando el IDLE Arduino v1.5.8, el cual sirve de traductor de lenguaje C a ensamblador de microcontrolador y se ejecuta en el microcontrolador.

A continuación se describe una serie de fases que se siguieron con sus respectivos resultados, para entender mejor el set de pruebas subsiguiente, no se relacionan todos los valores probados solo los últimos después de ajuste:

Implementación de algoritmo de conexión Cliente-Servidor:

Implementación de servidor:

Para la implementación del módulo del servidor en Python, se tomó como referencia un script de otro autor. Dicho script contenía la funcionalidad básica de comunicación por sockets en dicho lenguaje, además ciertas funciones para envío y recepción de comandos.

Implementación de cliente:

Al implementar el módulo del cliente en Arduino, se utilizó la documentación existente por parte de la plataforma, la cual define los protocolos previamente establecidos para la conexión del Arduino en modo cliente, el mismo se conecta al servidor recibiendo una dirección IP y un puerto al cual se desea enlazar. Dicho protocolo se define en el lenguaje C. En primer lugar el Arduino debe conectarse a la misma red del servidor, para lo cual e brindan los credenciales. Consecuentemente se crea un objeto del tipo WifiClient y se conecta al servidor creado en Python. Si todo resulta exitoso, la conexión está lista para enviar y recibir datos.

Determinación de tamaños aptos de transferencia:

Para la implementación de las siguientes pruebas se usó un RaspBerry PI como servidor y se utilizó un Arduino UNO R3 como cliente de la conexión, las pruebas iniciales se realizaron a un metro de distancia del Enrutador de conexión.

A continuación se procede a describir cada una de las pruebas realizadas y los problemas presentados a lo largo de la implementación y realización de las mismas.

Prueba básica:

Inicialmente se comenzó la implementación de las pruebas del protocolo WiFi utilizando la implementación documentada en la página oficial de Arduino, el código fuente se adaptó a las necesidades del este proyecto, en el cual se envían cadenas de bytes utilizando la conexión entre el cliente y el servidor.

Como se puede observar en las tablas, se enviaron ciertos patrones de bytes al cliente, y se midió el tiempo de respuesta del cliente, debido a esto, se intentó con cadenas pequeñas de 8, 16, 32, 64 y 128 bytes, sin embargo los envíos funcionaron correctamente a excepción de la cadena de 128 bytes, de la cual el servidor no recibió respuesta. A continuación en las tablas 30, 31 y 32, los tiempos de transferencia para cada uno de los patrones.

Tabla 30 Tiempos de transferencia de patrón 1

Patrón: 1					
Bytes	8	16	32	64	128
Tiempo	115.6568	145.5588	160.9117	207.3627	N/A
Tamaño:	8	16	32	64	0
Recibido:	✓	✓	✓	✓	x

Tabla 31 Tiempos de Transferencia de patrón 2

Patrón: 2					
Bytes	8	16	32	64	128
Tiempo	115.4509	145.3333	160.9215	207.5196	N/A
Tamaño:	8	16	32	64	0
Recibido:	✓	✓	✓	✓	x

Tabla 32 Tiempos de transferencia de patrón 3

Patrón: 3					
Bytes	8	16	32	64	128
Tiempo	114.6862	146.0098	160.7058	207.7843	N/A
Tamaño:	8	16	32	64	0
Estado:	✓	✓	✓	✓	✗

Prueba de opción alternativa de envío:

Basado en los resultados obtenidos es la sección anterior, se decidió buscar las posibles causas de error; considerando las limitadas opciones que brinda el protocolo de comunicación por red, se consideró como posible causa el tamaño del buffer de transferencia de la conexión, puesto que el cliente si recibió la cadena de bytes, pero el servidor no lo recibió de vuelta; sin embargo consultando fuentes no se pudo determinar el tamaño máximo permitido para el búfer, entonces se decidió escribir la cadena por fracciones de 64 bytes al socket y luego se imprime cambio de línea para que el socket envíe la cadena de bytes.

Adicionalmente se decidió agregar una función que mide la cantidad de memoria RAM disponible, dicho código se pudo obtener del sitio web oficial de Arduino, y se decidió medir la cantidad de bytes recibidos de vuelta por el servidor. Cada cadena debería ser 64 bytes más larga de su extensión original porque en el cliente a la cadena original se le agrega un identificador para saber si al llegar la cadena al servidor, se mantiene la integridad de la cadena original. Las tablas 33, 34 y 35 muestran el comportamiento obtenido.

Tabla 33 Segunda prueba con patrón 1

Patrón: 1				
Bytes	64	128	256	512
Tiempo	207.09803	297.2352	998.5784	1008.4117
Tamaño:	128	192	320	192
Recibido:	✓	✓	✓	✗

Tabla 34 Segunda prueba con patrón 2

Patrón: 2				
Bytes:	64	128	256	512
Tiempo:	205.0784	308.6078	1000.4411	1001.1862
Tamaño:	128	192	320	192
RAM:	830	676	210	858
Recibido:	✓	✓	✓	X

Tabla 35 Segunda prueba con patrón 3

Patrón: 3				
Bytes:	64	128	256	512
Tiempo:	219.8921	309.2843	1000.9901	1000.4607
Tamaño:	128	192	384	192
Recibido:	✓	✓	✓	x

Prueba con opción alternativa de almacenamiento:

Como se puede observar en las tablas 7, 8 y 9, los resultados fueron exitosos, los tamaños de envío de información hacia el cliente concuerdan con los tamaños de recepción desde el cliente, por lo cual se puede concluir que el algoritmo funciona correctamente para los tamaños de transferencia desde 1 byte hasta los 512 bytes. En las tablas 36, 37 y 38 se muestran los valores observados.

Tabla 36 Tercera prueba utilizando patrón 1

Patrón: 1				
Bytes	64	128	256	512
Tiempo	227.5690	393.9710	900.147	902.667
Tamaño:	128	192	320	576
Recibido:	✓	✓	✓	✓

Tabla 37 Tercera prueba utilizando patrón 2

Patrón: 2				
Bytes:	64	128	256	512
Tiempo:	230.4220	426.1670	901.255	899.490
Tamaño:	128	192	320	576
Recibido:	✓	✓	✓	✓

Tabla 38 Tercera prueba utilizando patrón 3

Patrón: 3				
Bytes:	64	128	256	512
Tiempo:	258.8730	390.6960	902.8240	898.3820
Tamaño:	128	192	320	576
Recibido:	✓	✓	✓	✓

Determinación de distancia apta para transferencia:

Después de corroborar el correcto funcionamiento del protocolo para varios dispositivos, se decidió comprobar el funcionamiento de la conexión, colocando el cliente a cierta distancia de servidor y además se desea aumentar la cantidad de clientes conectados simultáneamente para corroborar el correcto funcionamiento de la conexión.

Conexión de tres Arduinos a Raspberry PI a 10 metros del servidor:

Se analizará el comportamiento de 1, 2 y 3 Arduinos conectados a un Raspberry PI y enviándose datos simultáneamente a cada uno, comenzando inicialmente a una distancia de 2 metros desde el enrutador que se encuentra contiguo al servidor. En los datos presentes en la tabla la fila de Tiempo 1, Tiempo 2 y Tiempo 3 corresponden a uno, dos y tres Arduinos conectados respectivamente. Las tablas 39, 40 y 41 plasman los valores medidos.

Tabla 39 Tiempos de transferencia de patrón 1 a 2 metros del punto de acceso

Patrón: 1				
Bytes	64	128	256	512
Tiempo 1	227,569	393,971	900,147	902,667
Tiempo 2	457,991	820,138	1801,402	1802,157
Tiempo 3	716,864	1210,834	2704,226	2700,539
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

Tabla 40 Tiempo de transferencia de patrón 2, a 2 metros del punto de acceso

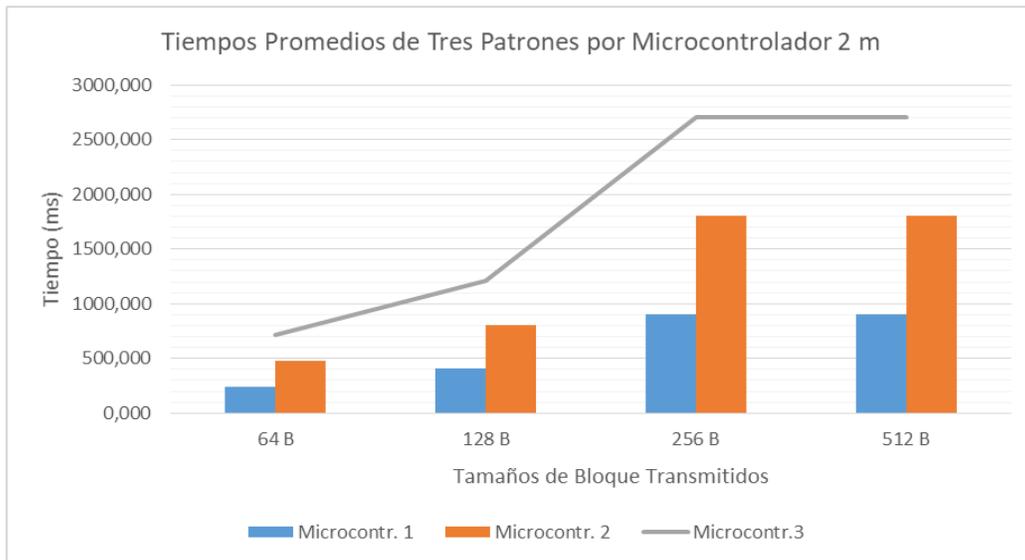
Patrón: 2				
Bytes	64	128	256	512
Tiempo 1	230,422	426,167	901,255	899,490
Tiempo 2	489,295	816,863	1804,079	1797,872
Tiempo 3	748,168	1207,559	2706,903	2696,254
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

Tabla 41 Tiempo de transferencia de patrón 3, a 2 metros del punto de acceso

Patrón: 3				
Bytes	64	128	256	512
Tiempo 1	258,873	390,696	902,824	898,382
Tiempo 2	486,442	784,667	1802,971	1801,049
Tiempo 3	685,560	1214,109	2701,549	2704,824
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

El gráfico 7 muestra los tiempos promedios de los diferentes patrones a una distancia de dos metros. Es de notar que la diferencia en tiempos entre 256B y 512B no es muy alta a pesar de que se dobla el tamaño de bloque transmitido.

Gráfico 7 Tiempos Promedios de Tres Patrones por Microcontrolador 2m



Conexión de tres Arduinos a PCDuino V2 a 10 metros del servidor:

Se analizará el comportamiento de 1, 2 y 3 Arduinos conectados a un PCDuino v2 y enviándose datos simultáneamente a cada uno, comenzando inicialmente a una distancia de 10 metros desde el enrutador que se encuentra contiguo al servidor. Las tablas 42,43 y 43 presentan los valores de tiempo para cada uno de los patrones con diferente tamaño.

Tabla 42 Tiempos de transferencia de patrón 1, a 11.85 metros del servidor

Patrón: 1				
Bytes	64	128	256	512
Tiempo 1	207,098	297,235	900,147	902,667
Tiempo 2	412,176	605,843	1801,402	1802,157
Tiempo 3	632,069	915,127	2704,226	2700,539
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

Tabla 43 Tiempo de transferencia de patrón 2 a 11.85 metros del servidor

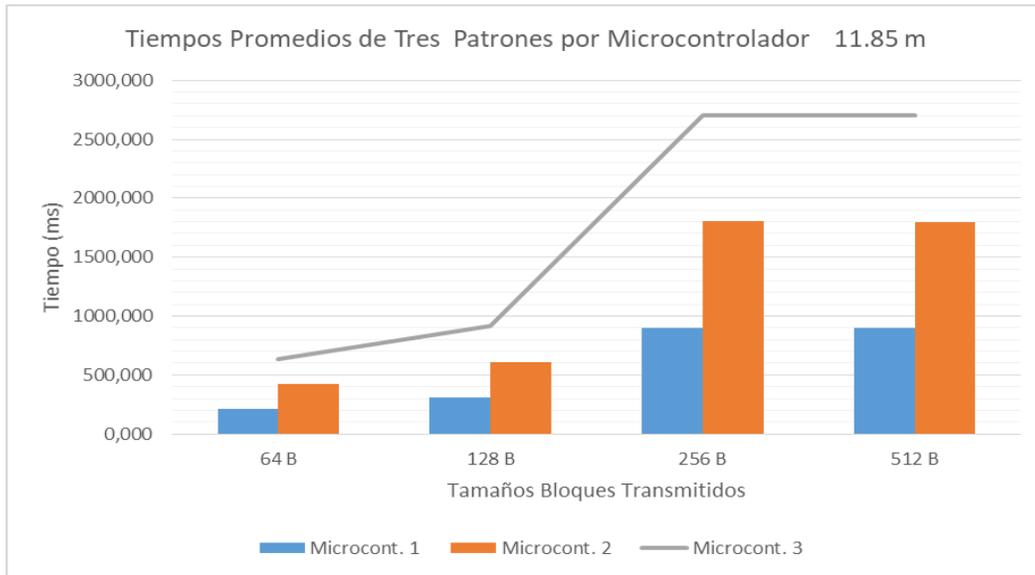
Patrón: 2				
Bytes	64	128	256	512
Tiempo 1	205,078	308,608	901,255	899,490
Tiempo 2	424,971	617,892	1804,079	1797,872
Tiempo 3	644,863	927,176	2706,903	2696,254
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

Tabla 44 Tiempo de transferencia de patrón 3 a 11.85 metros del servidor

Patrón: 3				
Bytes	64	128	256	512
Tiempo 1	219,892	309,284	902,824	898,382
Tiempo 2	426,990	606,520	1802,971	1801,049
Tiempo 3	619,274	903,078	2701,549	2704,824
Tamaño:	128	192	320	384
Recibido:	✓	✓	✓	✓

El gráfico 8 condensa los tiempos promedio para los patrones a una distancia de 11.85 metros. Se puede notar que el microprocesador 3, que fue el último en conectarse presenta un tiempo bastante alto.

Gráfico 8 Tiempos Promedios de Tres Patrones por Microcontrolador 11.85 m



Prueba de autonomía de módulo WIFI-SHIELD y módulo RF24:

Se realizaron pruebas de autonomía de ambas plataformas transfiriendo el Patrón 1 con 512 bytes de extensión entre un dispositivo y otro; dicha transferencia se realizó alimentando los módulos con broches de 4 baterías de 1,2v con 1200mA de capacidad cada una, luego se grabó un sketch específico para transferir constantemente cada cadena de un módulo a otro, finalmente se colocó los módulos en un sitio estable y se midió el tiempo desde que se comenzó la transferencia hasta que el dispositivo dejó de funcionar por falta de energía, además se midió las tensiones de la batería cuando comenzó la transferencia y cuando esta terminó.

A continuación se muestran los datos obtenidos y se especifica la tensión inicial de la batería al comenzar la transferencia, al finalizar y las corrientes mínimas y máximas alcanzadas durante ese tiempo de autonomía, el cual se especificó en minutos.

Prueba de autonomía RF24 se puede observar en la tabla 45:

Tabla 45 Mediciones de autonomía utilizando módulo RF24

Tensión Inicial (V)	Corriente Min. (mA)	Corriente Max. (mA)	Duración (minutos)	Tensión de Caída (V)
5,57	30,7	40,8	2151	5,01

Prueba de autonomía WIFI-SHIELD se puede observar en la tabla 46:

Tabla 46 Mediciones de autonomía utilizando módulo WiFi-Shield

Tensión Inicial (V)	Corriente Min. (mA)	Corriente Max. (mA)	Duración (minutos)	Tensión de Caída (V)
5,57	114,8	135,7	796	4,75

5.5 Cámaras

Con respecto a las cámaras se evaluaron varias y se preseleccionaron dos considerando principalmente costo y disponibilidad.

Las dos cámaras preseleccionadas se adquirieron y se les realizaron las pruebas de consumo y tiempos de transferencia. Con respecto a la Vimicro VC0706PREB los tiempos de transferencias logrados fueron tan lentos (superior a 3 minutos), que se descartó su uso para construir el módulo y por criterio de compatibilidad y tiempo se tomó la decisión de proseguir el desarrollo con la Raspberry Pi camera.

Resumen de Pruebas de Consumo con alimentación con baterías tamaño AA convencionales y recargables.

- Lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica.
- Toma de Mediciones: Asistente Sergio Vargas
- Descripción del lugar: Edificio A1-05 del Instituto Tecnológico de Costa Rica en el Laboratorio LuTEC, ubicado en segundo piso del edificio.
- Descripción del ambiente: Ambiente seco y climatizado a 23 grados Celsius.
- Fuente utilizada: USB

Tabla 47 Mediciones de consumo de corriente de Raspberry Pi B v1 en diferentes condiciones. Todos los valores se presentan en Amperios (A)

Placa Tricoder	Ocioso	Cámara	
		Local	Streaming
1	0,57	0,75	0,68
	0,57	0,76	0,68
	0,56	0,76	0,68
	0,56	0,75	0,68
	0,57	0,76	0,69
	0,56	0,76	0,69
2	0,56	0,75	0,63
	0,55	0,74	0,62
	0,56	0,75	0,63
	0,55	0,75	0,62
	0,55	0,74	0,62
	0,56	0,74	0,62
3	0,55	0,76	0,67

	0,57	0,76	0,68
	0,55	0,75	0,68
	0,56	0,74	0,67
	0,58	0,74	0,67
	0,55	0,75	0,68

Placa Tricoder	Ocioso	Cámara	
		Local	Streaming
4	0,56	0,74	0,67
	0,55	0,75	0,68
	0,55	0,75	0,67
	0,56	0,75	0,67
	0,55	0,74	0,67
	0,56	0,75	0,68
5	0,57	0,70	0,67
	0,56	0,71	0,67
	0,55	0,72	0,68
	0,56	0,71	0,67
	0,55	0,70	0,67
	0,56	0,71	0,67

Gráfico 9 Corriente Demandada por la Cámara con diferentes programadores

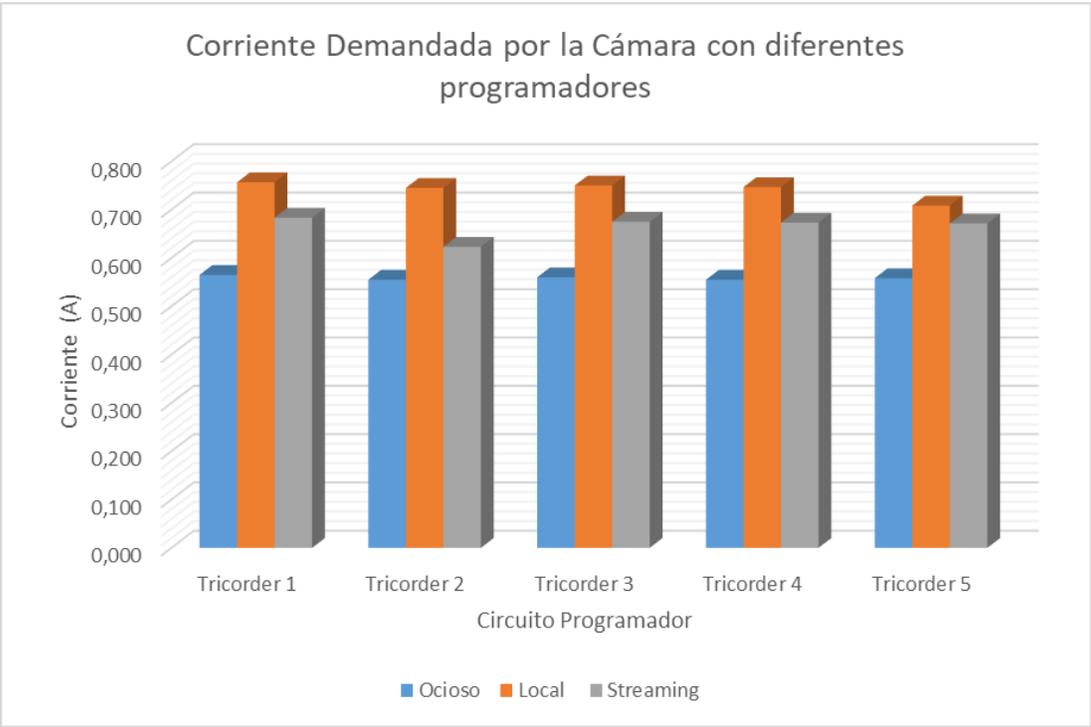


Figura 12 Cámara 1 probada Raspberry Pi, vista de frente



Figura 13 Cámara Vimicro, vista del lente

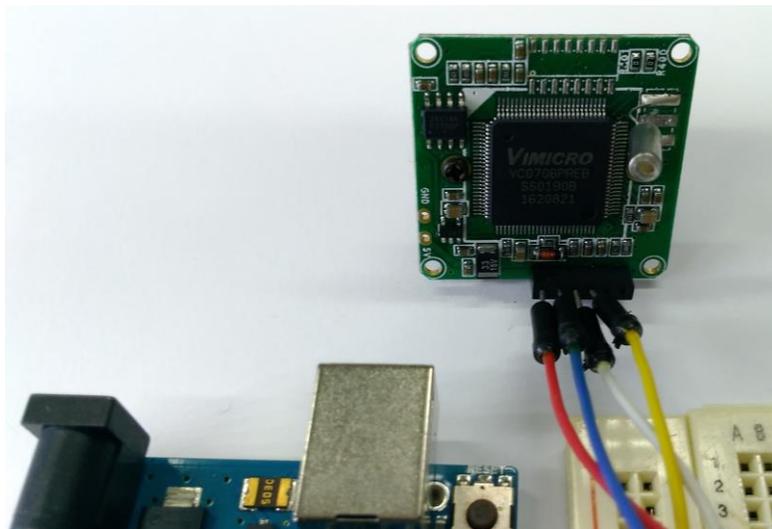


Figura 14 Cámara Vimicro, vista del circuito

A partir de los resultados de las evaluaciones preliminares, el estudiante de ingeniería electrónica desarrollo en (Calderón, 2015) propuestas para cámaras estereoscópicas, razonable para su uso en el Krotic.

5.6 Plan Piloto para Escuelas

Con un producto con este nivel de desarrollo ya se podría adaptar a diferentes situaciones y responder a muchas de las demandas de los chicos al aplicarlos en sus proyectos. De esta forma se plantea como parte del proyecto dos pruebas de campo extensivas. Estas serían en un estudio de evaluación con grupos de 16 a 20 chicos, para realizar pruebas en ambiente de aprendizaje real, usando los contactos generados en el pasado por el laboratorio LuTec del CIC. Posteriormente con esos datos se podría ya obtener los insumos para una actualización del robot. Con los planos diagramas y los códigos suministrados, si se mantienen los costos bajos, asociaciones de padres, juntas, o un ministerio, podrían disponer de este producto, para incorporarlo en sus programas de promoción de ferias científicas y las ferias de robótica que se verían pobladas por aplicaciones de robots no solamente basados el productos comerciales extranjeros sino en producción nacional.

A continuación se describe la guía elaborada, para el uso de los robots Krotic en el aula:

Guía para el uso de los Kits de Robótica Krotic en el aula

La dinámica para el uso del Krotic en el aula difieren a la mayoría de las dinámicas robots en el aula. En primera instancia el centro de la actividad no es el robot mismo sino sus módulos componentes. Otro aspecto es la programación del mismo, pues se realiza con dispositivo Se propio del Kit.

Se establece que el máximo número de estudiantes atender debe ser de 20 para 10 kits de robótica y 5 módulos de programación. Preferiblemente que al menos la mitad sean de género femenino. Las edades van de 11 a 15 años, esto quiere decir que ya están en capacidad de hacer abstracciones de la realidad.

Las dinámicas por proyectos necesitarán alrededor de 6 horas, preferiblemente distribuidos en dos días de un fin de semana.

Primera sesión 4 horas.

Segunda sesión, primera parte: 2 horas.

Segunda sesión, segunda parte: 2 horas

Requerimientos

Materiales de reuso: cartones, botellas plásticas, tapas plásticas, empaques tetrabick extendidos, pedazos de cable de red, juguetes viejos.

Materiales de acople: papel, témperas, lápices, compases, goma blanca, silicón, marcadores (tipo

pilot), madera balsa.

Materiales Krotic Módulos de sensores, Módulos del robot, motores DC, diodos rectificadores, transistores 2N2222, reguladores 5V, resistencias, LEDs.

Siete padrinos/madrinas

Aula sin pupitres con sillas para colocar en los lados.

Conexión eléctrica, fuente de agua y baño.

Habilidades que se promueven entre los participantes

Conocimientos

- Especialización: Fundamentos de física y electrónica funcionamiento de los módulos y algunos principios de electrónica básica. desarrollo de maquetas o construcción de prototipos como primera etapa de soluciones de problemas de la vida real.
-
- Administrativos: Preparación de la área de trabajo, zonas de construcción, preparación de zonas calientes de soldadura y silicón, cuidados reglas mínimas para preparación del área ocupacional administración de materiales para combinar materiales de reuso.
-
- Didácticos y pedagógicos:
 - Buscar información para establecer la especificación del proyecto.
 - Discernir cuál información se adapta.
 - Comunicación con los diferentes padrinos o madrinas y con los compañeros.
 - Aprender de la experiencia de lecciones impartidas

Actitudes

- Aprehensión,
- Tolerancia,
- interés,
- Liderazgo,
- Innovador,

- Ser retador / proactivo

Valores

- Respeto
- Responsabilidad
- Puntualidad
- Tolerancia
- Integridad
- honestidad
- estimular la creación de conocimientos

Plan didáctico de Sesiones:

En la primera sesión se establecen los grupos de trabajo.

Posteriormente los padrinos y madrinas presentan sus experiencias acerca del proceso de diseño de un producto, la importancia de construcción de los prototipos, destacar que todo el proceso es iterativo. Los padrinos resaltan también el hecho de que la primera aproximación no es la definitiva en la construcción de un producto.

Se definen los padrinos/madrinas, el proceso se sugiere con una dinámica y es preferible que sea aleatoria. Es importante evitar que los grupos sientan preferencias al momento de la asignación pues esto causaría desmotivación para el resto de la actividad.

Actividades de asignación de padrinos/madrinas con los grupos de trabajo

Especificación

Los participantes definen los proyectos del cual se construirán los prototipos.

Se plasman en las hojas los respectivos proyectos

Se hace una recolección de la información sobre las diferentes funcionalidades del proyecto, se procede con la especificación para definir el diseño.

Diseño

Se proponen formas de implantación e inclusive se le añaden o eliminan funciones al prototipo del producto. En esta etapa en especial hay una mayor libertad expresiva. Se plasma en las bitácoras.

Dentro de las pruebas se encontró que al finalizar este periodo era un momento adecuado para un receso, tomar un refrigerio y establecer un escenario de plática entre los padrinos/madrinas y los estudiantes participantes.

Con el diseño plasmado se hace una lista de materiales requeridos para la construcción.

Construcción

Se trabaja tanto en la apariencia del producto como en la funcionalidad. Para el aspecto funcional se requiere el apoyo de los padrinos/madrinas, quienes deben asegurarse de poder realizar los acoples electrónicos con la interfaz del robot y también la programación. Este aspecto es muy importante que los padrinos destaquen su papel de asesores no de constructores del dispositivo. Es por ello que las funcionalidades deben ser simples. No se debe subestimar el proceso de integración de los diferentes módulos del prototipo.

Se recomienda vehementemente a los padrinos/madrinas no asumir el papel de coautor/diseñador de los prototipos.

Normalmente a la mitad del proceso de construcción se finalizan las primeras 4 horas de trabajo.

La siguiente sesión se reinicia con la finalización del proceso de construcción. Se le da al producto el acabado necesario, se establece una estrategia de presentación del producto.

Acabado del Producto.

Al finalizar el proceso de acabado se realiza un receso para entrar en la última etapa de construcción del prototipo. La experiencia indica que este receso es bastante corto pues los estudiantes participantes están más interesados en finalizar el prototipo y preparar su presentación.

Presentación final de los prototipos de producto.

En esta etapa se realiza una sesión de presentación de las propuestas de cada uno de los grupos.

Se asigna un tiempo de presentación máximo, el recomendado es 5 minutos.

Un moderador introduce a cada grupo con los nombres de los integrantes y el proyecto.

Los grupos narran el proceso de especificación, diseño, construcción y acabado. Esta participación también permite a los padrinos y madrinan apoyar en la presentación.

Conclusión. Este es realizado por el o la moderadora, se hace una reflexión de los principales eventos sucedidos durante las 8 horas de la actividad.

Limpieza y cierre. Es importante dejar el local, al menos en las mismas condiciones en el que fue prestado. Despedida.

6 Discusión y Conclusiones

6.1 Quemador (Tricorder)

El diseño del Set de instrucciones, demostró ser lo suficientemente útil para probar las subrutinas típicas con algoritmos Subsumption de reacción de robot.

Para el quemador la batería B Batería Digital Energy (9.6V/2000mAh), demostró el tiempo de autonomía de 12.45 minutos de autonomía en proceso de grabación continua de 200 quemadas de programa, por lo que es una buena opción para adoptar como fuente de energía.

El Tricorder basado en la plataforma Raspberry Pi, demostró tener las herramientas necesarias para realizar las mediciones de las diferentes situaciones del circuito quemador, como para hacer transferencia de programa en código binario de otro Microcontrolador (crosscompiling).

El diseño con dos siete segmentos, posee más usabilidad que el de 1 despliegue aunque hace aumentar el costo del circuito en 4 dólares.

Se logró completamente el diseño y construcción del dispositivo quemador; más aún se realizaron mejoras sobre el diseño y se construyeron los dos módulos completos con otros tres módulos adicionales sin empaque.

6.2 Robot KROTIC

Los servomotores comerciales pueden presentar por fabricación o diseño, diferencias significativas de consumo hasta de un 400%. Para el caso de un robot con autonomía esto limita su tiempo de operación.

Para el caso del robot la fuente de alimentación con respecto a 9 V no presentó diferencia significativa (7.00%) en entre las recargables y las no recargables.

Para el caso de las baterías AA se determinó después de varias iteraciones (7), que lo mejor era añadir un par de baterías más, pues se está trabajando muy cerca del umbral del microcontrolador para operar en condiciones óptimas. Con esta nueva configuración se obtuvo solo una diferencia de tan solo el 14.28% entre la recargables y las convencionales o primarias.

En el caso de los algoritmos se encontró que las baterías de 9V entregaban más corriente al ejecutarse los tres algoritmos Subsumption de prueba. Por lo que se concluye

que en cuanto a corriente las baterías AA tanto recargables como las no recargables presentan mejor comportamiento.

En cuanto a consumo de potencia las baterías AA presentaron un comportamiento más parejo pues la diferencia fue apenas de 85 mW en cambio en las de 9 V hubo una diferencia de 600mW. Para el caso de las de 9V las baterías convencionales presentaron un mejor desempeño pues el consumo fue más bajo de 680 mW. Pero las AA presentan un comportamiento mucho más estable, por lo en términos de alimentación las baterías AA presentan una buena opción para trabajar durante una sesión de dos horas no continuas sin problema.

Se diseñó en forma iterativa el circuito del robot y su máquina virtual se llegó a un diseño funcional adecuado y se construyeron los diez prototipos en forma exitosa.

Se diseñó y construyó del compilador de la máquina virtual también se realizó usando el desarrollo iterativo de la adaptación del a metodología. Se construyó de forma exitosa aunque para efectos de eficiencia se puede mejorar.

Los programas Subsumption resultaron de menos de 3500 bytes de longitud, por lo que son económicos en términos de memoria.

6.3 Comunicación Inalámbrica

Como se puede observar, de los resultados obtenidos se muestreo que se pudo transmitir correctamente hasta 256KB sin fallas, al llegar a 512 KB, hubo fallas en la respuesta del servidor, pero además de eso, se verificó en el monitor del puerto serie del Arduino y se corroboró un tamaño de cadena recibido no correspondiente con el tamaño enviado desde el servidor. Además se pudo notar en los resultados que a partir de 256 kb, que existe un aumento en la memoria disponible, lo cual indica que podría haber una falla en la asignación de la memoria RAM del micro-controlador del Arduino, puesto que solo posee 2KB de RAM, probablemente la variable que almacena la cadena de bytes solo admite cierta cantidad de bytes y luego al llegar a un límite se resetea a su valor original.

6.3.1 Prueba con opción alternativa de almacenamiento:

Al observar un comportamiento anómalo en la asignación de la memoria RAM del microcontrolador, se analizó el código fuente del protocolo de transmisión y recepción de datos en C, y se detectó que la variable que almacena la lectura que viene del servidor es de tipo String, con lo cual se puede suponer de esta situación, una causa justificable del gran

uso de memoria por parte del cliente, como solución alternativa se decidió implementar el almacenamiento de la cadena de bytes recibidos en un arreglo de tipo char.

Las variables tipo char, almacenan un solo byte de información, con lo cual se asegura que cada byte que llega desde el servidor, ocupará un solo byte en la memoria RAM del cliente, luego que se almacenó la cadena recibida, se procedió a escribir por partes hacia el socket del servidor y finalmente enviar la cadena final hacia el mismo.

Como se puede observar en las tablas y gráficos los resultados fueron exitosos, los tamaños de envío de información hacia el cliente concuerdan con los tamaños de recepción desde el cliente, por lo cual se puede concluir que el algoritmo funciona correctamente para los tamaños de transferencia desde 1 byte hasta los 512 bytes.

Para las pruebas realizadas el Módulo de transferencias resultó ser más efectivo en cuanto a tiempo de transferencia que el de WiFi de Arduino, lo que indica que es el mejor para construir el módulo basado en el circuito RF24.

En cuanto a la distancia de comunicación el módulo RF24 mostró un menor tiempo con de respuesta con respecto al módulo de WiFi .

Cuando se probó la transmisión con varios microcontroladores receptores se notó un incremento de 340% en bloques de 256 y 512 bytes esto quiere decir que el último microcontrolador en conectarse tardará mucho más en recibir la información del emisor. No se considera un problema grande pues los programas no resultan ser muy grandes.

6.4 Plan Piloto para Escuelas

Implementar el plan dentro del aula, requirió reuniones preliminares con los estudiantes que servirían de padrinos/madrinas, esto para familiarizarlos con: los módulos, el robot, sus componentes, además de especializar a algunos en funciones de observación y medición. La dinámica se desarrolló sin problemas. Por lo que se concluye, que está bien estructurada y puede ser escalable, para otras escuelas en las mismas condiciones limitadas recursos.

7 Recomendaciones

Se ofrecen las recomendaciones conducentes a la continuación del trabajo con el robot Krotic, que por el desarrollo de las tecnologías de comunicación, se ven bastante promisorias en especial la de IoT.

7.1 Programador

Se recomienda optimizar consumo en circuitería de interfaz, desactivar componentes de hardware y software que consumen energía. También evaluar la posibilidad de cambiar de plataforma (Raspberry Pi modelos Zero, B+, A+).

Considerando el avance de plataformas en velocidad y funcionalidad de los circuitos es necesario trabajar en el rediseño del conjunto de instrucciones, así como el software general sistema incluyendo el compilador, considerando especialmente de mantenibilidad del software. Uso de ensamblador en combinación con C/C++ para ejecución más eficiente, y tener mayor control sobre el hardware

También se hace necesario para la operación diaria agregar algún tipo de indicador de estados de error y un procedimiento de recuperación de ese estado.

Agregar hardware de administración de energía con soporte para rutinas de arranque y finalización seguros, así como re inicio del sistema.

7.2 Set de Instrucciones

En vista de la deficiencia detectada en instrucciones condicionales y ciclos; existen condicionales, pero no consideran todas las posibilidades se sugiere extender el conjunto de instrucciones o agregar soporte para prefijos. También se sugiere evaluar la incorporación de instrucciones para operaciones a nivel de bit, lógicas y multiplicación.

Considerar opciones para la ejecución de tareas de forma simultánea: tales como instrucciones con función específica de alto nivel por ejemplo moverse hasta evento, o directamente un formato de instrucciones con soporte para multitasking, ejecución de loops en conjunto con otra tarea.

Agregar instrucción de sleep para estado de bajo consumo.

Establecer claramente las instrucciones no ortogonales, de esta forma se pueden definir cuáles modos de direccionamiento no serán válidos para una instrucción específica.

Mejorar documentación con respecto a pines, registros y sus funciones (hoja de datos gráfica).

7.3 Krotic (Hardware)

Trabajar con baterías AA ya sea recargables o convencionales.

Verificar cada servomotor a utilizar ya que por defecto o desajuste puede consumir potencia más allá del promedio.

Agregar indicador de estado (LED), para señalar el estado del robot en un momento determinado.

Cambiar de regulador a uno con menor tensión de dropout que se ajuste a la tensión de las baterías. Incrementar la tensión de entrada de las baterías. Agregar convertidor boost.

Actualmente la rotación de servos es imprecisa debido a que la rotación por ángulo involucra rotar por un tiempo determinado. Se sugiere considerar un modo de calibración, para modificar el tiempo de rotación, de acuerdo con la tensión de la batería.

Se recomienda la Impresión en 3D de diferentes piezas de más piezas que las actuales.

Módulo inalámbrico.

Continuar con el desarrollo de una tecnología equivalente o más actualizada del RF24 de Nordic.

Agregar al set de instrucciones, instrucciones propias de transferencia usando los módulos inalámbricos.

7.4 Plan Piloto para Escuelas

Aunque se cuenta con experiencia para aplicar los robots en el aula con un enfoque constructivista, se recomienda continuar la labor de investigación en este aspecto, pues se considera tan o más importante que el desarrollo del robot mismo.

8 Referencias

- Agile Coaching, A. M. (11 de 11 de 2013). *Agile Methodology*. Obtenido de Agile Methodology: <http://agilemethodology.org/>
- Brooks, R. A. (2000). *Intelligence Without Reason. MIT Press. Invited Papers*, 569-595.
- Calderón, C. A. (2015). *Detección y Evación de Obstáculos para una Plataforma Robótica Educativa (KROTIC)*. Cartago.
- Dimitris Alimisis, M. M. (2006). *Robotics & Constructivism in Education: the TERECoP project*. European Programme Socrates/Comenius/Action 2.1, Agreement No 128959-CP-1-2006-1-GR COMENIUS-C21 2006 – 2518 / 001 – 001 SO2.
- Grupo Estrategia Siglo XXI. (2006). *Estrategía Siglo XXI: Conocimiento e Innovación hacia el 2050 en Costa Rica*. San José, Costa Rica.: Fundación Costa Rica Estados Unidos de América para la Cooperación.
- Morales, J. M. (2014). *Desarrollo e Implementación de una Red de Comunicación Inalámbrica para el Proyecto de Investigación KROTIC*. Cartago.
- Papert, S. (1993). *The Children's Machine, Rethinking School in the age of the Computer*. Basicbooks.
- Russell Knight, F. F. (2001). *Balancing Deliberation and Reaction, Planning and Execution for Space Robotic Applications*. *International Conference on Intelligent Robots and Systems* (págs. 2131-2139). Maui, Hawaii: Proceedings of the 2001 IEEE/RSJ.
- Stallings, W. (2013). *Computer Organization and Architecture Designig for Performance* . New Jersey, NY.: Pearson Education, Prentice Hall.
- Stuart J. Russell, P. N. (2010). *Artificial Intelligence A Moderm Approach* . New Jersey 07458, NY.: Pearson Education, Inc.
- Tanembaum, A. (2006). *Structured computer organization* . USA: Pearson Prentice Hall.
- Tucker Balch, J. S. (2008). *Designing Personal Robots for Education: Hardware, Software, and Curriculum*. *IEEE Pervasive Computing Volume 7 Issue 2* .
- Valverde, B. R. (2014). *Desarrollo de Módulo de Audio para Transmisión y Recepción Inalámbrica*. Cartago.

Villegas-Lemus, M. (2010). *Informe Final Krotic*. Cartago.

Yousuf, M. A. (2009). *Robots in Education*. Santa Fe, México.