



“SISTEMA PARA EL MONITOREO, DETECCIÓN Y ALERTA DE SOMNOLENCIA DEL CONDUCTOR MEDIANTE VISIÓN ARTIFICIAL, COMUNICACIÓN INALÁMBRICA Y GEOLOCALIZACIÓN”

Segundo Reporte Parcial

Lista de actividades

- Diseño del Módulo Central de procesamiento
- Diseño del Submódulo de Procesamiento de Imágenes
- Diseño de la unidad contenedora del Módulo central de Procesamiento
- Búsqueda de Lenguajes de Programación para el Microordenador
- Investigación de Estándares y Protocolos de Comunicación Inalámbricas
- Búsqueda de Interfaces de Comunicación Inalámbrica
- Elección de Interfaces de Comunicación Inalámbrica
- Búsqueda de Tecnologías de Geolocalización

Autores:

Alan Eduardo Gamboa Del
Ángel
Maite Paulette Díaz Martínez
Nilsen Alfaro Parra

Asesores:

M.en C. Niels Henrik Navarrete
Manzanilla
Dr. Rodolfo Vera Amaro

Índice

1. Diseño del Módulo Central de Procesamiento	4
2. Diseño del Submódulo de Procesamiento de Imágenes	5
3. Diseño de la Unidad Contenedora del Módulo Central de Procesamiento	6
4. Búsqueda de Lenguajes de Programación para el Microordenador	7
5. Investigación de Estándares y Protocolos de Comunicación Inalámbricas	10
6. Búsqueda de Interfaces de Comunicación Inalámbrica	13
7. Elección de Interfaces de Comunicación Inalámbrica	16
8. Búsqueda de Tecnologías de Geolocalización	18
9. Conclusiones	19
10.Bibliografia	20
11.Apéndice A: Códigos fuente	21
12.Apéndice B: Planos de manufactura	22
13.Apéndice C: Datasheets	22
14.Apéndice D: Piezas físicas	22

Índice de figuras

1.	Aplicaciones industriales del robot SCARA	5
2.	Programa clásico vs Machine Learning	7
3.	Tipos de Redes Inalámbricas	11
4.	Mapa de la Ciudad de México	11
5.	Mapa de Cobertura LTE-Telcel	16
6.	Base del eslabón 3	22
7.	Émbolo del eslabón 3	22
8.	Volantes del eslabón 3	23
9.	Bielas y tapa del eslabón 3	23
10.	Soporte para motor y elemento rotatorio del eslabón 3	24
11.	Soporte del actuador del antebrazo	24
12.	Soporte para el encoder del antebrazo	25

Índice de tablas

1. Diseño del Módulo Central de Procesamiento

El presente trabajo reporta el proceso de diseño e implementación del proyecto titulado “Diseño, manufactura y control de un robot paletizador articulado de cuatro grados de libertad asistido por un sistema de visión artificial” como Trabajo Terminal para que cada uno de los integrantes del equipo obtenga el título de Ingeniero en Mecatrónica en la Unidad Profesional Interdisciplinaria en Ingenierías y Tecnologías Avanzadas (UPIITA).

Este Trabajo Terminal consiste en el diseño y construcción de un sistema para la clasificación y paletización de llaves como proceso posterior al troquelado de estas en distintas líneas de producción. En esta entrega final del reporte se detalla el proceso de diseño y re-diseño de los elementos y subsistemas pertenecientes al proyecto. El sistema se divide en varios subsistemas, siendo los más importantes el subsistema de clasificación, que contará con una interfaz en SimpleCV¹ para la clasificación de llaves utilizando técnicas de visión artificial, el transporte de las llaves se basa en una banda transportadora y el subsistema de paletización para el que se diseñará y constuirá un robot SCARA de 4 grados de libertad controlado por computadora.

El sistema planteado debe ser capaz de operar de manera autónoma y continua con un mínimo de errores, el usuario será capaz de modificar la velocidad de trabajo y cargar algunas rutinas de tiempos, esto debido a que se plantea construir un sistema flexible, que pueda ser fácilmente adaptado a tareas diversas.

La primera parte de este escrito consiste en los antecedentes necesarios y el planteamiento del problema así como la metodología utilizada para la selección de la mejor solución. Posteriormente se describirá el diseño y modelado de los subsistemas que integran la solución escogida. Se reportan así mismo las simulaciones correspondientes al control propuesto para el subsistema de paletizado, los análisis de esfuerzos en la estructura mecánica, así como los resultados experimentales de la integración completa del sistema de manera virtual.

La última parte del documento corresponde a la conclusión del diseño y re-diseño de elementos y evidencias de implementación del sistema

¹Software de código abierto para el desarrollo de sistemas de visión artificial

2. Diseño del Submódulo de Procesamiento de Imágenes

El presente trabajo integra las cuatro disciplinas de la mecatrónica para generar un sistema que ayude a eficientar procesos repetitivos, se plantea un primer acercamiento a la optimización del proceso antes mencionado, sentar las bases para futuros trabajos a realizarse en la UPIITA, así como para fomentar nuevas aplicaciones y mejoras a este prototipo.

Tomando como base los fundamentos educativos adquiridos durante nuestra estancia en la Unidad y la experiencia de nuestros asesores, se plantea buscar la mejor solución y con ello las herramientas y procesos necesarios para lograrla.

En el año 2005 en la UPIITA se realizó un trabajo parecido a nuestro proyecto en el sentido de presentar un brazo robótico, este para realizar prueba de trayectorias en aplicaciones didácticas.

En el ámbito de la visión artificial, se han realizado diversos proyectos, entre los cuales destacan un sistema de visión artificial para la navegación de un robot móvil (2005), un dispositivo automatizado para el control de posición de una marcadora mediante visión artificial (2009) y un prototipo de un roboconstructor autónomo con visión artificial para el concurso EUROBOT 2009.

El proyecto está pensado para ser lo más económico posible para poder implementarse en pequeña y mediana industria, ya que los modelos comerciales oscilan entre los 21,000 y 50,600 euros.

Se espera sentar un precedente en cuanto a proyectos que tengan como objetivos la clasificación y paletización de objetos.

A continuación se muestra un robot SCARA en una aplicación industrial:



Figura 1: Aplicaciones industriales del robot SCARA

3. Diseño de la Unidad Contenedora del Módulo Central de Procesamiento

El desarrollo económico de un país está directamente ligado a la independencia tecnológica de éste, en un país emergente como México, es de suma importancia la investigación, el diseño y la construcción de prototipos que propongan mejoras a artefactos y optimización de procesos industriales.

El diseño e implementación de tecnología en la industria e investigación agiliza los procesos, convirtiéndolos en automáticos o autónomos y garantizando la precisión de estos.

El proceso de paletización es requerido en todos los niveles de la industria, en ocasiones como proceso intermedio o bien de almacenaje, en específico para la industria mediana se realiza esta tarea de forma manual. Usualmente al proceso de paletizado lo antecede un proceso de separación o clasificación de elementos, este por requerir el discernimiento de forma, color, materiales, etc es comúnmente realizado de por una persona o grupo de personas.

El proceso de fabricación de llaves para cerradura consiste básicamente en uno o varios sistemas de troquelado seriados con un proceso de paletizado, que a su vez está seriado con un proceso de control numérico. Usualmente los procesos de clasificación y paletizado son realizados de forma manual, lo que acarrea retardos e incremento de costos.

Para el continuo funcionamiento, la precisión y la reducción de costos en la fabricación de llaves de cerradura se propone el diseño y construcción de un sistema integral, compuesto por un sistema de visión artificial para la clasificación de las llaves, una banda transportadora y un robot SCARA de 4 grados de libertad para el proceso de paletizado.

4. Búsqueda de Lenguajes de Programación para el Microordenador

Lo primero que debemos de tener claro, es que la programación orientada a la inteligencia artificial es diferente al paradigma de la programación convencional. En esta última, el usuario le indica a la máquina exactamente lo que tiene que hacer, mientras que en Machine learning, se le enseña a programarse sola. Lo cual se ejemplifica en el siguiente gráfico:

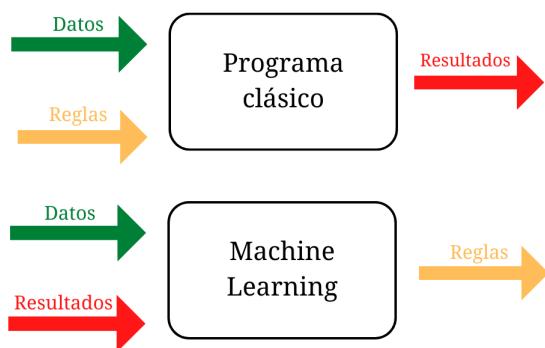


Figura 2: Programa clásico vs Machine Learning

El proceso de trabajo para aprendizaje automático es muy diferente a la construcción de una aplicación convencional. Por este motivo, la manera de utilizar los lenguajes de programación es diferente. Se deben de tomar en cuenta las características que estos utilizan, así como sus enfoques y paradigmas.

Uno de los factores importantes a considerar al momento de elegir un lenguaje de programación orientado a machine learning, es la popularidad el mismo, ya que esta es una señal de la aceptación por parte de la comunidad. A su vez, su ecosistema es tanto o mas importante, ya que podemos darnos una idea si dicho lenguaje posee las herramientas adecuadas que se acoplan a nuestras necesidades. Finalmente, la ideonidad del lenguaje es otro factor relevante, ya que, si el lenguaje fue diseñado con una determinada tarea o propósito en mente, este será mucho más eficiente y productivo.

A continuación, se listan algunos de los lenguajes de programación más populares en el campo de machine learning:

- **Python**

Python se ha convertido en el lenguaje de programación más popular del mundo, esto gracias

a su sencillez aunado a una gran versatilidad. Sus utilidades abarcan desde desarrollo web, hasta la inteligencia artificial. Python es uno de los lenguajes más naturales y con sintaxis menos complicada. Por lo tanto, es fácil de aprender, leer y corregir errores. Lo mejor de todo es que Python es un lenguaje de código abierto. Lo que significa que es de libre acceso y distribución. Como resultado, tiene bibliotecas versátiles.

Una biblioteca es una colección de módulos o códigos creados para realizar operaciones específicas. Algunos módulos pueden ayudar con la interfaz gráfica de usuario, análisis de datos, gráficos, entre otras cosas. En lugar de escribir el mismo código cada vez que lo necesite, puede importarlo desde la biblioteca.

Uno de los principales factores de esta popularidad, es el enorme ecosistema de herramientas desarrolladas específicamente para machine learning tales como: Pandas, Numpy, Tensorflow, Keras, entre otras. A pesar de no ser el lenguaje más rápido o potente, lo anterior lo hace uno de los lenguajes indispensables si se piensa trabajar en el campo de machine learning.

- **C++**

A menudo hemos escuchado que C++ se aborda más cuando se trata de desarrollar juegos y sistemas grandes. Esto se debe a su característica de portabilidad y también proporciona una comprensión básica de la construcción lógica. Sin embargo, Algunos de los paquetes compatibles con C++ incluyen: Microsoft Cognitive Toolkit (CNTK) para aprendizaje profundo; Tensorflow para aprendizaje profundo; OpenCV para visión artificial; MLPack para aprendizaje automático; DyNet para redes neuronales; OpenNN para redes neuronales; Shogun para aprendizaje automático y FANN para redes neuronales.

Sin embargo, C++ también tiene sus propias deficiencias. Está muy orientado a la sintaxis, a diferencia de Python, que es realmente amigable para los principiantes. Ademas de no contar con un gran soporte de bibliotecas.

- **R**

Otro de los lenguajes de programación cuyo uso es considerablemente popular en el ámbito de machine learning y data análisis es R. R ofrece una variedad de beneficios desde su buen soporte de biblioteca y gráficos hasta la posibilidad crecer enormemente con muchos de los que podemos buscar ayuda, debido a su naturaleza abierta. Algunos de los paquetes clave compatibles con R son: Kernlab para operaciones basadas en regresión y clasificación; Ggplot para visualización de datos; Caret para operaciones basadas en regresión y clasificación; Plotly para visualización de datos; MLr3 para diferentes flujos de trabajo de aprendizaje automático. al igual que Python, también es comparativamente más lento que el resto de los lenguajes. Esto representa una gran desventaja para este lenguaje de programación. Además de esto, también tiene un origen débil, por lo que no es tan fácil de aprender. Para las personas que no tienen experiencia en programación les puede resultar difícil aprender R.

- **Matlab**

Matlab también admite operaciones de aprendizaje automático y se puede aplicar de manera innovadora a aplicaciones que incluyen visión por computadora. MATLAB no está limitado por la sintaxis y, por lo tanto, es fácil de aprender y comprender. MATLAB también tiene muchas funciones predefinidas y una GUI para que los alumnos entiendan mejor las cosas. Además, viene con un compilador MATLAB que ayuda cuando se trata de codificar en el mismo. Además, cuando se trata de capacidades de aprendizaje automático, MATLAB lo admite de una manera única. MATLAB permite a los usuarios aplicar AutoML, lo que les permite aprovechar al máximo los modelos codificados optimizados y reducidos. Además, MATLAB también puede realizar la generación automática de código para el análisis de sensores y mucho más. MATLAB también viene con algunas restricciones. El lenguaje de programación no está fácilmente disponible o es de uso gratuito. De hecho, una versión de prueba está disponible solo por 30 días y publica que los usuarios deben comprar el paquete para un uso posterior. Por esta razón, uno puede encontrar solo investigadores como su público objetivo. No solo eso, el compilador de MATLAB es costoso de comprar y, por lo tanto, no tiene un soporte de comunidad / foro denso para ayudar a los usuarios y estudiantes.

- **Javascript**

JavaScript no será tu lenguaje principal para desarrollo en Inteligencia Artificial, pero sí que es un gran complemento para hacer muchas cosas, especialmente visualización, y conviene aprenderlo porque te va a servir para todo un poco. Si quieras exponer tus modelos hacia el exterior, seguramente lo acabarás haciendo a través de alguna interfaz web. El hecho de que TensorFlow.js se ejecute dentro del navegador abre una gama de posibilidades interesantes tanto para empresas como para desarrolladores. Como los navegadores son un espacio interactivo: ofrece acceso a varios sensores, incluidas cámaras web y micrófonos, que pueden proporcionar imágenes y sonidos como entrada a cualquier modelo de aprendizaje automático. A pesar de todos los aspectos positivos, TensorFlow.js no tiene acceso predeterminado al sistema de archivos en el entorno del host del navegador. Esto limita los recursos de datos disponibles y puede imponer restricciones en el tamaño de los archivos.

5. Investigación de Estándares y Protocolos de Comunicación Inalámbricas

Para transferir datos o información de un punto a otro sin la utilización de cableado o algún medio físico, tenemos las redes inalámbricas; y se dividen en 3 diferentes tipos dependiendo del alcance requerido:

- **La red de área personal inalámbrica (WPAN)**

Son redes de corto alcance utilizando tecnología Bluetooth. Interconectan dispositivos compatibles como teléfonos cerca de una ubicación central. Son conocidas como redes de corto alcance, con cobertura de 10 a 100 metros

- **La red de área local inalámbrica (WLAN)**

Viene como resultado de las señales de teléfonos móviles proporcionadas por los proveedores de servicios móviles. Son conocidas como redes de corto/mediano alcance alcanzando de 100 a 1000 metros.

- **La red de área personal local inalámbrica (WNAN)**

Son conocidas como redes de corto/mediano alcance llegando de 5 a 10 km.

- **La red de inalámbrica de area amplia (WWAN)**

Usan ondas de radio pero transmite a uno o varios puntos de acceso inalámbrico donde un usuario inalámbrico puede conectarse a la red.

En esta red se incluye:

- **LPWAN(Low Power Wide Area Network): Red de Área Amplia de Baja Frecuencia**

Son conocidas como redes de largo alcance con cobertura de hasta 100km

A continuación, se muestra de manera gráfica los tipos de redes antes mencionadas, así como algunos de sus protocolos y tecnologías.

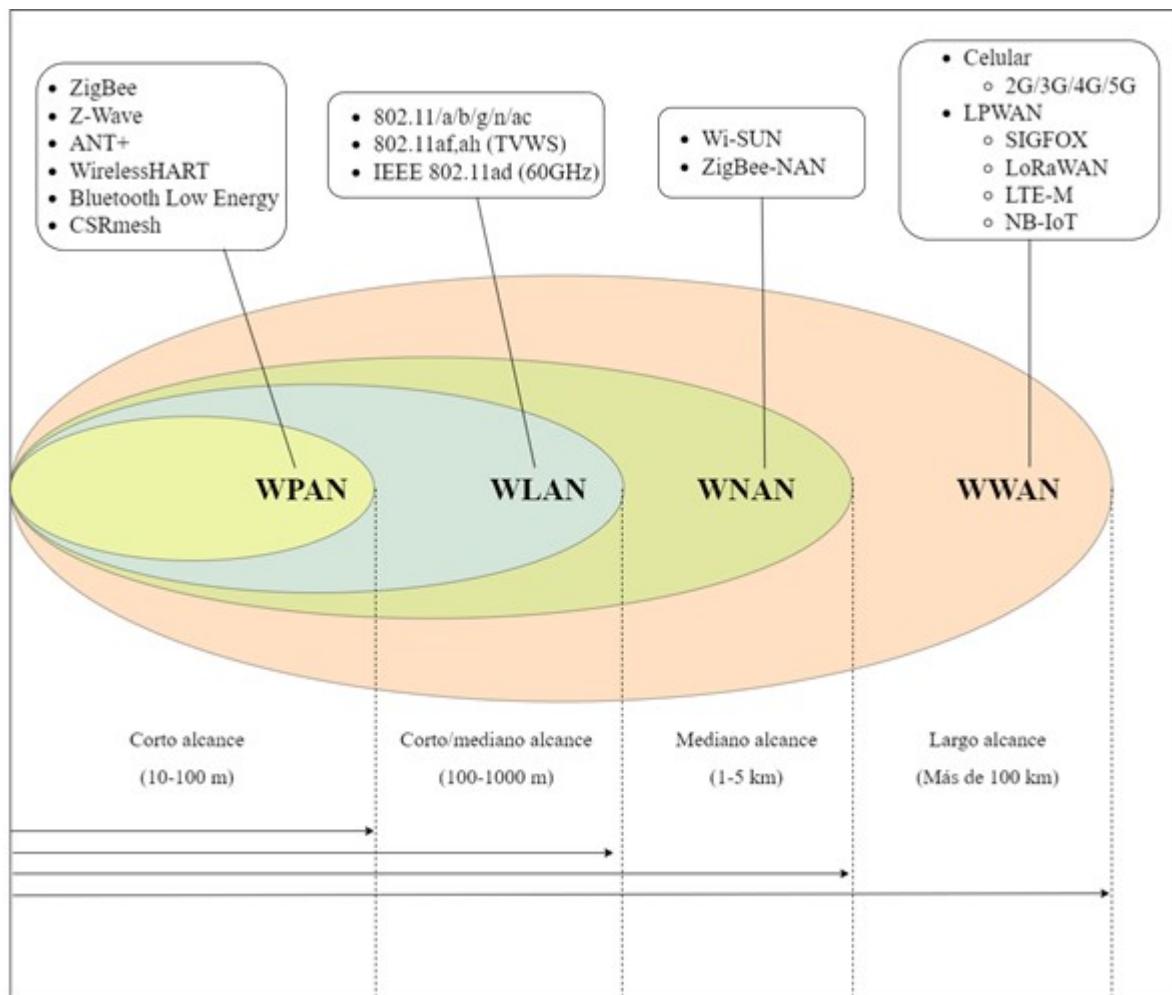


Figura 3: Tipos de Redes Inalámbricas

Dado que para el presente proyecto tiene contemplado funcionar dentro de la Ciudad de México y considerando que esta tiene una extensión de 1485 km², el tipo de red mas viable a utilizar es la LPWAN.



Figura 4: Mapa de la Ciudad de México

A continuación, se muestra un cuadro comparativo con estos tipos de redes.

	Espectro	Ancho de Banda	Costo	BitRate	Escalabilidad	Inmunidad a la Interferencia
LTE-M	Sin licencia 869 MHz 915 MHz	100kHz	Moderado	1Mbps	Alta	Moderada
NB-IoT	Sin licencia 433 MHz	<500kHz	Moderado	200kbps	Alta	Baja
SigFox	Con licencia 700-900 MHz	200kHz	Bajo	100bps	Baja	Baja
LoraWan	Con licencia 700-900 MHz	1.4MHz	Bajo	10kbps	Moderada	Muy Alta

Ya que una parte fundamental del proyecto es la transferencia de contenido multimedia, se requiere un bitrate mínimo de 0.5 Mbps.

Tomando esto en consideración, podemos observar que tanto SigFox y LoraWan no cumplen con este requisito, por lo tanto quedan descartadas para ser utilizadas.

Por lo tanto, las tecnologías que cumplen estas características son LTE-M y NB-IoT, por lo que en la siguiente sección se procede a hacer un análisis y comparación de estas dos tipos de redes inalámbricas para corroborar cual de las dos es la más óptima para ser utilizada en este proyecto.

6. Búsqueda de Interfaces de Comunicación Inalámbrica

LTE-M LTE-M es el término simplificado de la industria para el estándar tecnológico de Área Amplia de Baja Potencia (LPWA por sus siglas en inglés). Se refiere específicamente a la tecnología LTE CatM1, que es el estándar más adecuado para su uso con el IoT.

Es una tecnología de área amplia de baja potencia que admite IoT a través de dispositivos poco complejos y proporciona una cobertura extendida, al tiempo que permite la reutilización de la base instalada de LT

Las redes LTE-M coexistirán con las redes móviles 2G, 3G y 4G y se beneficiarán de todas las características de seguridad y privacidad de las redes móviles, como la confidencialidad de la identidad del usuario, la autenticación de la entidad, la privacidad, la integridad de los datos y la identificación del equipo móvil.

A continuación se describen las ventajas principales de LTE-M

■ Ahorro de Energía

LTE-M permite que los dispositivos IoT usen un modo de ahorro de energía, que les permite ir a dormir cuando no están en uso. Y tampoco tienen que despertarse para enviar solo la actualización de ubicación, sino que pueden extender los tiempos de suspensión por un período de tiempo más largo. LTE-M también permite que los dispositivos usen recepción discontinua extendida (eDRX). Cuando el dispositivo está fuera del modo suspendido, verifica periódicamente la información del enlace descendente. Con eDRX, se aumenta el tiempo de verificación de la radio, lo que reduce el consumo de energía.

■ Cobertura

Las redes LTE-M/Cat-M1/Cat-M2 pueden utilizar la infraestructura 4G LTE existente. Esta es una ventaja significativa, ya que más del 50 por ciento de las conexiones móviles globales se realizan en redes 4G.

Con una pérdida de acoplamiento máxima (MCL) de 156 decibelios (dB), 14 dB más alta que LTE, las redes LTE-M también ofrecen mayor cobertura y mejor penetración en interiores. La señal puede manejar muchas interferencias de edificios y otras estructuras que obstruyen su camino.

■ Velocidad de Transferencia

Comparado con LTE, LTE-M no es particularmente rápido. Pero 1 megabit por segundo para transmisiones de enlace ascendente y descendente es excepcional en comparación con redes más antiguas como 2G y 3G (UMTS) y otras LPWAN. (Es varias veces más rápido que NB-IoT). Y eso es más que suficiente para la mayoría de las aplicaciones de IoT. Es factible para aplicaciones con necesidades de datos pequeñas e incluso casos de uso que involucran transmisión de video.

En el contexto de IoT, el alto rendimiento de datos significa que los dispositivos que usan LTE-M pueden recibir fácilmente actualizaciones por aire (OTA), y las transmisiones de datos consumirán menos energía, porque el dispositivo puede volver al modo de ahorro de energía más rápido.

■ **Costo**

Las redes 4G se construyeron principalmente para teléfonos inteligentes. Las redes LTE-M se construyeron principalmente para dispositivos IoT. Como tal, los componentes necesarios para los dispositivos LTE-M son menos complejos y más asequibles que los componentes que necesitaría para un dispositivo 4G tradicional, aunque ambos utilizan la infraestructura 4G LTE.

NB-IoT

Es un protocolo inalámbrico de Internet de las cosas (IoT) que utiliza tecnología de red de área amplia de baja potencia (LPWAN). Fue desarrollado por 3GPP para comunicación inalámbrica celular que permite una amplia gama de nuevos dispositivos y servicios NB-IoT. NB-IoT es uno de los tres principales estándares 3GPP LPWAN.

El estándar de comunicación NB-IoT tiene como objetivo permitir que los dispositivos IoT funcionen a través de redes de operador, ya sea dentro de una onda portadora de comunicación del Sistema Global para Móviles (GSM) existente, en una "banda de protección" no utilizada entre canales LTE o de forma independiente.

Uno de los objetivos de NB-IoT es impulsar la extensión de la cobertura más allá de lo que ofrecen las tecnologías celulares existentes. Para ello, NB-IoT ofrece repeticiones de transmisión y diferentes configuraciones de asignación de ancho de banda en la transmisión de enlace ascendente.

La tecnología NB-IoT utiliza señales de bajo ancho de banda para comunicarse dentro de las tecnologías GSM y LTE existentes.

Los dispositivos y sensores especialmente diseñados son los componentes básicos de los sistemas NB-IoT. Estos dispositivos recopilan información de su entorno y la transmiten a estaciones base NB-IoT o nodos de transmisión.

Las estaciones base individuales están conectadas a una puerta de enlace de IoT y servidores de aplicaciones en la nube de IoT para el monitoreo centralizado y el análisis de datos.

NB-IoT emplea una nueva capa física con señales y canales para cumplir con los requisitos de cobertura extendida en áreas rurales e interiores profundos, al tiempo que permite una complejidad de dispositivo muy baja. La tecnología subyacente es mucho menos compleja que la de los módulos GSM/GPRS.

A continuación se describen las ventajas más importantes de NB-IoT:

■ **Ahorro de Energía**

NB-IoT no necesita ejecutar un sistema operativo pesado, como Linux, ni hacer mucho procesamiento de señales, lo que lo hace más eficiente en términos de energía en comparación con otras tecnologías celulares.

- **Cobertura**

NB-IoT puede ayudar a admitir una gran cantidad de dispositivos mediante el establecimiento de redes NB-IoT que pueden conectarse a miles de millones de nodos. Diseñado para cobertura extendida en interiores, la menor complejidad de los dispositivos proporciona conectividad y comunicación de largo alcance.

- **Costo**

Debido a que es más fácil crear dispositivos con menor complejidad, el costo de los dispositivos es significativamente bajo, alrededor de 5 dólares por módulo.

- **Seguridad**

NB-IoT está protegido de manera muy similar a 4G, incluidas todas las funciones de autenticación basadas en cifrado y SIM.

7. Elección de Interfaces de Comunicación Inalámbrica

LTE-M vs. NB-IoT

Comparado a NB-IoT las velocidades de transferencia de datos de LTE-M son más de 10 veces más rápidas, su latencia es de 10 a 100 veces menor y cuenta con una cobertura mucho mayor, ya que utiliza la infraestructura 4G LTE existente. Sin embargo, aunque ambas LPWAN funcionan bien en interiores, NB-IoT tiene una pérdida máxima de acoplamiento (MCL) ligeramente más alta, lo que significa que puede manejar un poco más de interferencia.

LTE-M también puede usar una gama más amplia de frecuencias, aunque, como su nombre lo indica, Narrowband-IoT usa bandas de frecuencia más estrechas, lo que permite que esta tecnología use el espectro de radiofrecuencia (RF) de manera más eficiente.

	Velocidad Máxima de Subida	Velocidad Máxima de Descarga	Latencia	Perdida Máxima de Acoplamiento(dBs)	Ancho de Banda
LTE-M	1Mbps	1Mps	10-15 ms	156	1.4MHz a 5MHz
NB-IoT	127 kbit	159 kbit	1.6-10s	164	180KHz

Después de la investigación realizada, encontramos las siguientes ventajas de LTE-M sobre NB-IoT:

- LTE-M tiene más redes de roaming disponibles

Pocos operadores han establecido acuerdos de roaming para sus redes NB-IoT, por lo que LTE-M ofrece una mejor cobertura y tiene menos circunstancias en las que necesita cambiar de SIM. Actualmente, la compañía Telcel cuenta con la mayor cobertura de tecnología LTE en México.

Figura 5: Mapa de Cobertura LTE-Telcel

- NB-IoT consume más energía en algunos casos

Cuando los dispositivos necesitan transmitir grandes volúmenes de datos, las velocidades de datos más bajas de NB-IoT requieren estos permanezcan en línea por más tiempo, lo que resulta en un mayor consumo de energía. En casos de uso móvil, NB-IoT también usa más energía

Por lo anterior, se decidió a utilizar el estándar LTE-M para nuestro proyecto.

Dado que el presente proyecto será desarrollado para funcionar en una Raspberry Pi 4 modelo B, procederemos a comparar distintos módulos compatibles para Raspberry Pi que nos permita utilizar y conectarlos a la red LTE-M

Modelo	Voltaje de Operación	Redes Soportadas	Dificultad de Integración	Costo
SixFab Base Hat	5V	3G/4G/LTE	Bajo	\$2500
SIM7600A-H	3-5V	3G/4G/LTE	Bajo	\$1436
SIM800C-G	3.3 - 5V	3G/4G/LTE	Alta	\$508
SIM7600CE	3.3-5V	2G/3G/4G/LTE	Bajo	\$2134

8. Búsqueda de Tecnologías de Geolocalización

9. Conclusiones

10. Bibliografía

Referencias

- [1] Rafael Kelly, Victor Santibáñez Davila, Antonio Loría, *Control of Robot Manipulators in Joint Space*, Segunda edición ,Springer, 2005
 - [2] Mark W. Spong, Seth Hutchinson y M. Vidyasagar, *Robot Modeling and Control*, primera edición, JOHN WILEY & SONS, INC, 2005.
 - [3] Antonio Barrientos, Luís Felipe Peñinm Carlos Balaguer, Rafael Aracil , *Fundamentos de robótica*, Primera edición , McGraw-Hill, 1997
 - [4] Joseph E Shigley, John J Uicker Jr, *Teoria de Máquinas y Mecanismos*, Primera edicicón , McGraw-Hill, 1998
 - [5] Robert L. Mott, *Diseño de elementos de máquinas*, Cuarta edición , Pearson, 2006
 - [6] Javier Echávarri Otero, Giuseppe Carbone, Marco Ceccarelli, José Luis Muñoz Sanz, *CRITERIOS PARA LA SEGURIDAD EN EL USO DE ROBOTS*, 8º CONGRESO IBEROAMERICANO DE INGENIERIA MECANICA, 2007
 - [7] Kurt Demaagd, Anthony Oliver, Nathan Oostendorp y Katherine Scott *Practical Computer Vision with SimpleCV*, Primera edición, O'REILLY, 2012
 - [8] A. Serrano Nicolás *Neumática Práctica*, Primera edición, Paraninfo , 2009
 - [9] Gareth J. Monkman, Stefan Hesse, Ralf Steinmann, Henrik Schunk *Robot Grippers*, Primera edición, Wiley-VCH , 2007
- Cibergraffía:
- http://web.ing.puc.cl/~mtorrest/downloads/robotics_tutorial/pres_robots_summer_school_111214.pdf
- <http://www.pdcahome.com/1932/qfd-despliegue-calidad/>

11. Apéndice A: Códigos fuente

Como se mencionó en la sección de simulaciones, las primeras simulaciones fueron realizadas sin el modelo importado de *SolidWorks* a *SimMechanics*, se utilizó en su lugar una bloque de Simulik llamado planta para cada eslabón del manipulador robótico.

A continuación se presenta el código correspondiente al bloque de planta del eslabón 1:

12. Apéndice B: Planos de manufactura

13. Apéndice C: Datasheets

A continuación se presentan las hojas de especificación de los motores, elementos de transmisión (engranes, poleas, bandas) y sensores (encoders).

14. Apéndice D: Piezas físicas

En esta sección se muestran las piezas físicas con las que se cuentan hasta el momento en tanto al eslabón 3 y el antebrazo.

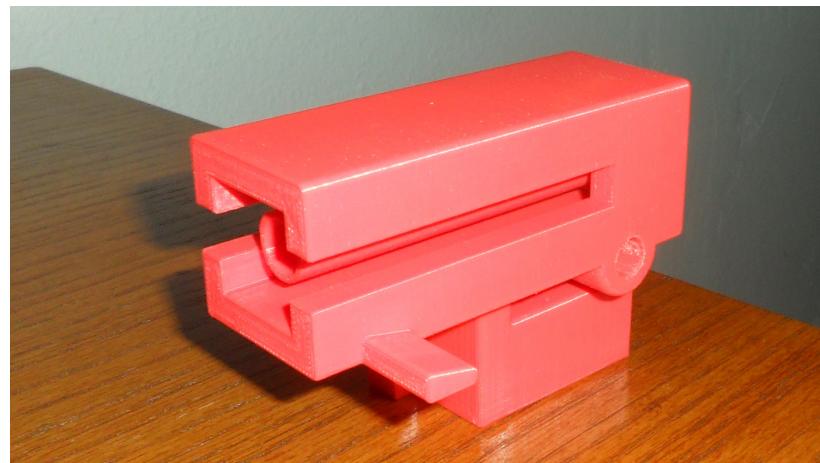


Figura 6: Base del eslabón 3

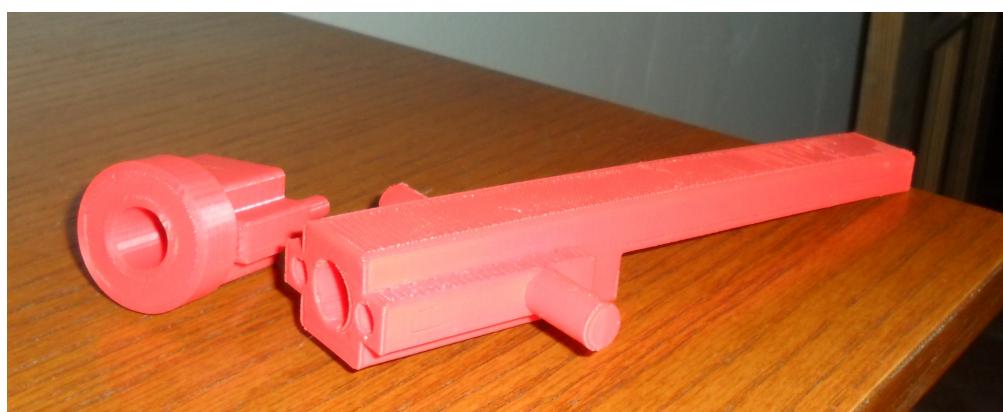


Figura 7: Émbolo del eslabón 3



Figura 8: Volantes del eslabón 3



Figura 9: Bielas y tapa del eslabón 3

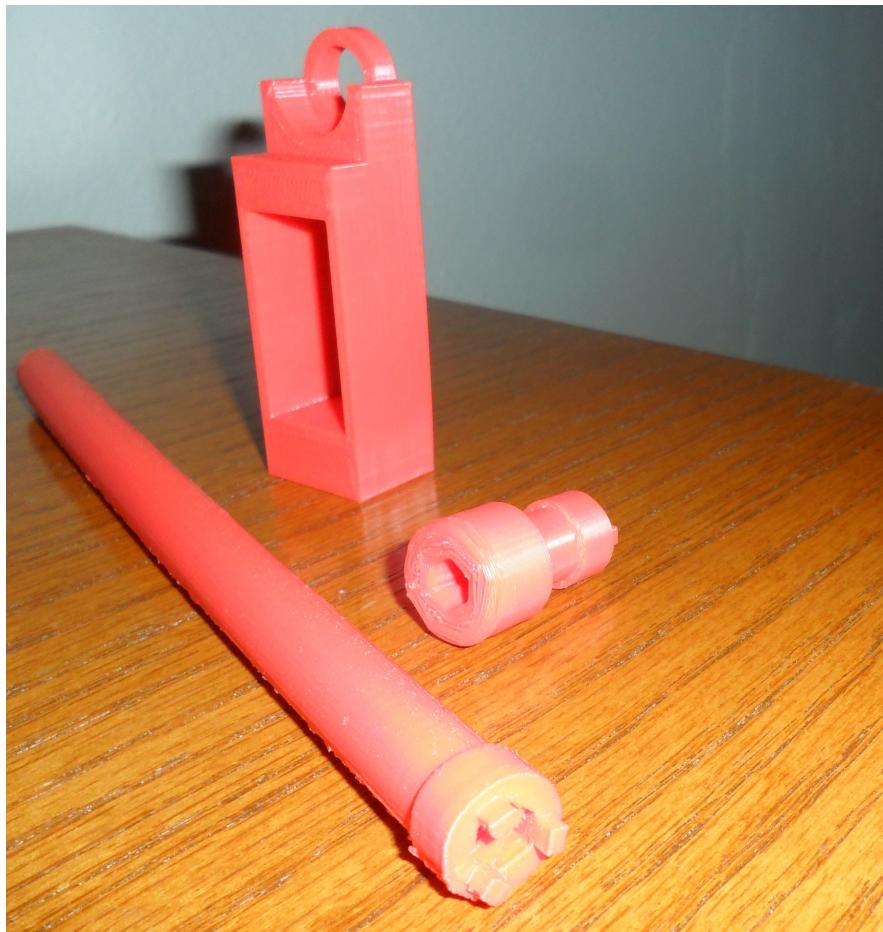


Figura 10: Soporte para motor y elemento rotatorio del eslabón 3

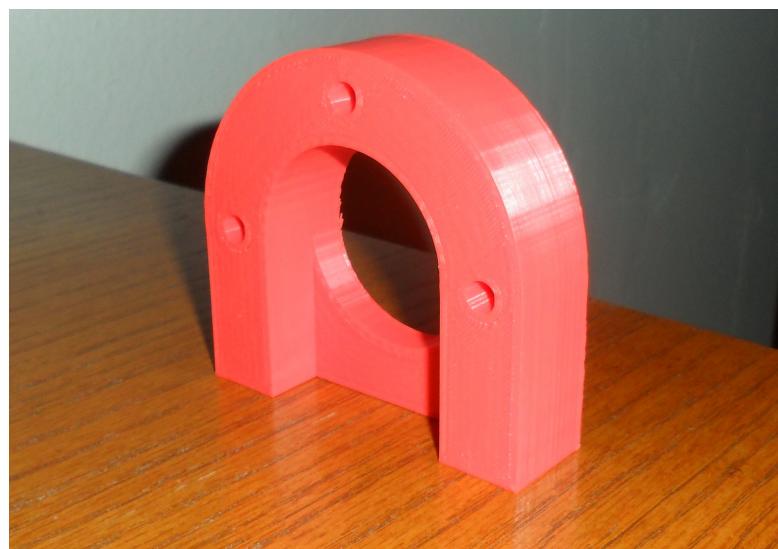


Figura 11: Soporte del actuador del antebrazo

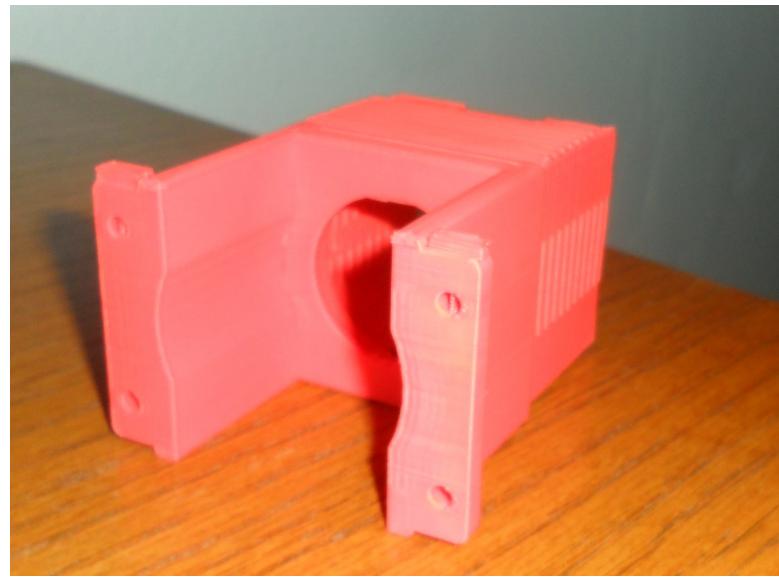


Figura 12: Soporte para el encoder del antebrazo

Ro-
 bot
 $F = m \frac{d\dot{x}}{dt}$ grados de li-
 bertad IPN Visión artificial
 $\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i$ $x_{c1} = \ell_{c1} \operatorname{sen}(q_1)$
 IPN $A_n^0 = A_1^0 A_2^1 \dots A_n^{n-1}$ SCARA
 $\phi = \arctan \left(\frac{p_y}{p_x} \right)$ Mecatrónica
 $k_2 = \frac{1}{2} m_2 v_2^2 + \frac{1}{2} I_2 (\dot{q}_1^2 + \dot{q}_2^2)$
 Dinámica inversa $\mathcal{L} =$
 $k - u \sum_i^k F_i^T \delta r_i = 0$
 upiita
 ♡

$$\ddot{e} + K_d \dot{e} + K_p e = 0$$

$$\tau = M(q)V_{aux} + C(q, \dot{q})\dot{q} + G(q)$$