

CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR



Colegio de Ingeniería  
Dirección de Posgrado  
Campus Mexicali

Tesis/Proyecto de Ingeniería e Innovación

**Desarrollo de Módulo Adaptador IoT para Monitoreo y Control de Señales en  
Sistemas que Carecen de Comunicación Inalámbrica.**

para obtener el grado de  
Maestro en Ingeniería e Innovación

Presenta

**Kevin Bennett Ruiz López**

**Director de proyecto de tesis:** Dr. Luis Carlos Básaca Preciado

**Codirector de proyecto de tesis:** Dra. Verónica Rojas Mendizábal

**Codirector de proyecto de tesis:** Dr. Moisés Jesús Castro Toscano

Mexicali, Baja California. Septiembre, 2020

CENTRO DE ENSEÑANZA TÉCNICA Y SUPERIOR



Colegio de Ingeniería  
Dirección de Posgrado  
Campus Mexicali

Tesis/Proyecto de Ingeniería e Innovación

**Desarrollo de Módulo Adaptador IoT para Monitoreo y Control de Señales en  
Sistemas que Carecen de Comunicación Inalámbrica.**

para obtener el grado de  
Maestro en Ingeniería e Innovación

Presenta

**Kevin Bennett Ruiz López**

**Comité Evaluador:**

---

**Dr. Miguel A. Ponce Camacho**

**Dr. Juan M. Terrazas Gaynor**

**Dr. Jorge M. Jauregui Sesma**

Mexicali, Baja California. Septiembre, 2020

## **Agradecimiento y dedicatoria.**

L.B.

M.C.

V.R.

K.G.

R.B.

P.R.

K.R.

E.N.

E.W.

D.R.

B.O.

A.O.

L.T.

A.V.

L.H.

O.R.

Gracias al apoyo de todas esas personas este trabajo pudo finalizarse, y de corazón muchas gracias por apoyarme en momentos difíciles que todos ellos saben que tuve que pasar para llegar hoy y sin ellos ni comenzado estuviera este trabajo.

Este trabajo se pudo realizar gracias al apoyo del Centro de Innovación y Diseño (CEID), Coordinación Institucional de Investigación de CETYS Universidad y los recursos del Consejo Nacional de Ciencia y Tecnología (CONACYT).

**Carta Institucional.**



**Mexicali, Baja California a 11 de septiembre de 2020**

**Atención a**

**Dra. Dania Licea Verduzco.**

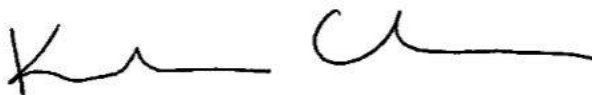
Coordinadora de Ingeniería en Diseño Gráfico Digital.

Colegio de ingeniería, Campus Mexicali.

Por medio de la presente se hace constar que el proyecto de aplicación con el título “Desarrollo de Módulo Adaptador IoT para Monitoreo y Control de Señales en Sistemas que Carecen de Comunicación Inalámbrica.” Presentado por el alumno Kevin Bennett Ruiz López con matrícula 029113, en el programa de Maestría en Ingeniería e Innovación, se desarrolló exitosamente en su estancia en el programa en CETYS Universidad campus Mexicali, bajo la dirección académica de los doctores, Luis Carlos Básaca Preciado, Verónica Rojas Mendizábal y Moisés Jesús Castro Toscano docentes de la escuela de ingeniería.

Se extiende la presente para los fines que al interesado convenga.

**ATENTAMENTE**



---

**Dra. Karla Garduño Palomino**

Coordinadora Académica de Posgrado de Ingeniería Campus Mexicali.

## Índice General.

Agradecimiento y dedicatoria.....	iii
Carta Institucional.....	iv
Índice de Figuras.....	vii
Índice de Tablas.....	viii
Lista de abreviaturas.....	ix
Resumen.....	x
Capítulo 1. Introducción.....	1
1.1 Antecedentes.....	1
1.3 Planteamiento del Problema.....	4
1.4 Preguntas de investigación.....	5
1.5 Objetivos.....	6
1.5.1 Objetivo General.....	6
1.5.2 Objetivos Específicos.....	6
Capítulo 2. Marco Teórico.....	7
2.1 Internet de las cosas.....	7
2.1.1 Identificación por radio frecuencia (RFID).....	7
2.1.2 Red de sensores inalámbricos (WSN).....	8
2.1.3 Middleware.....	9
2.1.4 Cómputo en la nube.....	10
2.1.5 Software de aplicación IoT.....	11
2.2 Aplicaciones IoT áreas Industrial y Comercial.....	13
2.2.1 Industria 4.0.....	13
2.2.2 IoT Aplicaciones Comerciales.....	15
2.2.3 Vehículo autónomo CETYS Universidad.....	19
2.3 Dispositivos Adaptadores.....	20
2.3.1 Comerciales.....	20
2.3.2 Desarrollo Investigación.....	22
2.4 Dispositivos / Protocolos / Capa de Aplicación.....	23
2.4.1 Dispositivos.....	23
2.4.1.1 National Instruments roboRIO.....	24
2.4.1.2 ATmega 2560 microcontrolador.....	24

2.4.1.3 ATmega 328 microcontrolador. ....	24
2.4.1.4 WeMo D1 mini microcontrolador. ....	25
2.4.1.5 Raspberry Pi 3.....	25
2.4.1.6 Victor SPX (controladores de motores). ....	25
2.4.1.7 Optical Encoders. ....	26
2.4.1.8 Sensores Ultrasónicos.....	26
2.4.2 Protocolos.....	26
2.4.2.1 I2C.....	26
2.4.2.2 SPI.....	27
2.4.2.3 UART.....	27
2.4.2.4 CAN.....	27
2.4.3 Capa de Aplicación.....	28
2.4.3.1 HTTP .....	28
2.4.3.2 MQTT .....	28
2.5 Servidores en la nube. ....	29
3.1 Primera Etapa. ....	32
3.2 Segunda Etapa. ....	34
3.3 Tercera Etapa. ....	36
Capítulo 4. Resultados. ....	38
4.1 Primera Etapa .....	38
4.2 Segunda Etapa .....	39
4.3 Tercera Etapa .....	44
Capítulo 5. Conclusiones. ....	48
Referencias. ....	51
ANEXOS.....	58

## Índice de Figuras.

Fig. 1. Vista simplificada de transferencia de datos en una RFID tag pasiva de baja frecuencia. ....	8
Fig. 2.a Arquitectura de red WIRELESSHEART para aplicación de Industria 4.0. Fig. 2.b Arquitectura de red ISA100 para aplicación de Industria 4.0. ....	9
Fig. 3. Arquitectura de software de una plataforma de servicio de Industria 4.0. ....	10
Fig. 4. Aplicaciones y funciones de cómputo en la nube. ....	11
Fig. 5. Actividades de IoT en el Comercio. ....	16
Fig. 6. Aplicaciones de IoT en edificios inteligentes. ....	17
Fig. 7. Diagrama de flujo para las pruebas de los componentes internos del vehículo. ....	32
Fig. 8. Diagrama de flujo para las pruebas de las nubes y protocolos de la capa de aplicación. ....	34
Fig. 9. Diagrama de flujo para el módulo adaptador dentro del vehículo. ....	37
Fig. 10. Diagrama de conexión para la prueba del protocolo HTTP. ....	40
Fig. 11. Diagrama de conexión para la prueba del protocolo MQTT. ....	41
Fig. 12. Diagrama de conexión para la prueba para nube de AWS con el protocolo de comunicación MQTT. ....	42
Fig. 13. Diagrama de conexión para la prueba para nube de IBM Cloud con el protocolo de comunicación MQTT. ....	42
Fig. 14. Diagrama de conexión para la prueba para nube de Node-RED con el protocolo de comunicación MQTT. ....	43
Fig. 15. Captura de pantalla del tablero interactivo con datos del vehículo autónomo. ....	45
Fig. 16. Diagrama de conexión para la prueba del módulo adaptador para la comunicación de los componentes internos del vehículo autónomo a la nube de Node-RED. ....	46
Fig. 17. Diagrama Esquemático del Módulo Adaptador. ....	47

## Índice de Tablas.

Tabla 1. Resultados de los protocolos alámbricos por cada dispositivo interno del vehículo. ....	38
Tabla 2. Resultados de las pruebas por cada protocolo de comunicación alámbrica. .	39
Tabla 3. Resultados de los protocolos de la capa de aplicación por cada nube. ....	40
Tabla 4. Resultados de los protocolos de comunicación de la capa de aplicación. ....	41
Tabla 5. Resultados de las pruebas nubes comerciales y locales. ....	43
Tabla 6. Resultados de la prueba del módulo adaptador con la nube Node-RED. ....	44



## **Lista de abreviaturas.**

**AWS** (*Amazon Web Services*) Servicios Web Amazon

**CAN** (*Controller Area Network*) Controlador de Red de Zona

**CNC** (*Computer Numerical Control*) Control Numérico por Computadora

**CPPS** (*Cyber-Physical Production Systems*) Sistemas de Producción Ciber Físicos

**CPS** (*Cyber-Physical Systems*) Sistemas Ciber Físicos

**HTTP** (*Hyper Text Transport Protocol*) Protocolo de Transferencia de HiperTextos

**I2C** (*Inter-Integrated Circuit*) Circuito Inter-Integrado

**I4.0** (*Industry 4.0*) Industria 4.0

**IoT** (*Internet of Things*) Internet de las cosas

**MQTT** (*Message Queuing Telemetry Transport*) Transporte de Telemetría de Cola de Mensajes

**PLC** (*Programmable Logic Controller*) Control Lógico Programable

**RFID** (*Radio Frequency Identification*) Identificación por Radio Frecuencia

**SPI** (*Serial Peripheral Interface*) Interfaz Periférica Serial

**UART** (*Universal Asynchronous Receiver and Transceiver*) Transmisor-Receptor  
Asíncrono Universal

**WSN** (*Wireless Sensor Networks*) Red de Sensores Inalámbricos

## Resumen.

En CETYS Universidad campus Mexicali, se trabaja en un proyecto de investigación que consiste en el desarrollo de un vehículo autónomo eléctrico para el transporte de la comunidad CETYS dentro del mismo campus. El cual ha tenido varias versiones de prototipos, que han provocado cambios a la arquitectura que se había planteado inicialmente, y con ello componentes se han removido y otros se agregaron, llegando a un punto en que se necesita un dispositivo que pueda comunicar el vehículo autónomo hacia al Internet y más específico aún a una nube, donde puede interactuar en ambas direcciones. Algunos de los dispositivos que integran el vehículo cuentan con módulos de comunicación W-Fi. El objetivo de este trabajo es el desarrollo de un módulo adaptador para poder enviar información a una plataforma digital donde se pueda realizar un monitoreo e interacción de datos y comandos a larga distancia con múltiples dispositivos, sensores y actuadores, trabajando en la integración de diferentes protocolos de comunicación por medio de servicios de nube e Internet de las cosas.

Para llegar a eso se utilizó una metodología de tres etapas, donde cada etapa aportará una o más características para Módulo Adaptador, además se obtendrán resultados que ayudarán a la siguiente etapa para la preparación de sus pruebas. En la primera etapa se enfoca en encontrar algún dispositivo ya interno del vehículo para funcionar como el módulo adaptador y el protocolo de comunicación alámbrico. En la segunda etapa su objetivo es especificar cuál es la nube que se utilizara, puede ser tanto privada, comercial o local, además de establecer el protocolo de comunicación de la capa de aplicación hacia la nube seleccionada. La última etapa se enfoca más en las pruebas ya con el dispositivo adaptador comunicándose con el resto de los dispositivos del vehículo y mando información a la nube para desplegarla en un tablero interactivo.

Uno de los resultados de este trabajo fue un módulo adaptador conformado por dos dispositivos (ATmega 2560 y Wemo D1 mini) los cuales ya se encontraban dentro del vehículo. Adicionalmente a eso, se redujo el número de dispositivos que se requerían para su operación, la mejor nube que se probó fue una local. La información fue recolectada, enviada y desplegada correctamente en el tablero interactivo, con velocidades de transferencia aceptables para los requerimientos del proyecto.

# Capítulo 1. Introducción.

## 1.1 Antecedentes.

Hoy en día, la mayoría de los dispositivos electrónicos utilizados en la vida diaria, en la industria y en otras áreas, tienden a estar conectadas a Internet e intercomunicados, se pronostica que para este año 2020, se tendrán alrededor de 30 miles de millones de dispositivos conectados a Internet (Nordrum, 2016). Dos grandes temas que han estado tomando fuerza en los últimos años son la Industria 4.0 (I4.0) e Internet de las cosas (IoT). Por lo tanto, estas tecnologías han impulsado diversas aplicaciones como monitoreo remoto, acceso rápido a la información, procesamiento en la nube, e incluso tomas de decisiones para sistemas autónomos (Hu et al., 2016). Sin embargo, hay muchos dispositivos electrónicos que siguen siendo competentes y utilizables que no tienen un modo de acceso a Internet, ¿Qué pasará con todos estos dispositivos? ¿Están destinados a ser obsoletos solo por el hecho que no son compatibles con estas nuevas tecnologías?

En la actualidad, el Internet de las cosas (IoT) es una red global de máquinas y dispositivos capaces de interaccionar unos con otros. El IoT es una de las áreas más importantes de las tecnologías futuras y está ganando una gran atención en un amplio rango de la industria (Gubbi et al., 2013). IoT ha generado un gran impacto en las empresas desde que los dispositivos están completamente conectados y comunicados entre sí, esto se ve en aplicaciones como en la integración de los sistemas de inventario, sistemas de atención al cliente, aplicaciones inteligentes para el negocio y análisis de negocios (Lee & Lee, 2015).

En la industria moderna se están presentando grandes cambios debido a su cuarta transformación, donde los dispositivos y robots que se usan dentro de la industria están siendo conectados a Internet para crear ecosistemas ciber-físicos. El futuro de las compañías deberá tener muy en cuenta esta tecnología por la relevancia e impacto que está teniendo para los clientes y proveedores; para los proveedores en la toma de tiempos de entregas de material y cuando solicitar material para la empresa evitando

retrasos en producción, en los clientes para ofrecer los productos, entregas y soporte personalizados a sus necesidades. Entonces, nuevas tecnologías y protocolos se están desarrollando para comunicaciones de los dispositivos y robots. Las nuevas tecnologías de IoT que se están desarrollando para ofrecer servicios de comunicación, análisis, toma de decisión, y sistemas complejos con compatibilidad a IoT. La industria requerirá innovar el proceso y el equipamiento que se estén usando en estos ecosistemas (Chen et al., 2018; Zhou et al., 2015).

Las redes de Sensores Inalámbricos (WSN, por sus siglas en inglés de *Wireless Sensor Network*), surgieron para comunicar dispositivos inalámbricos a corto alcance, y así crear redes inalámbricas, las cuales dieron entrada a IoT, debido que comunicándose por medio de protocolos inalámbricos con un adaptador a Internet podían crecer su rango de comunicación con otras redes o nodos a gran escala. Esta tecnología conecta diversos tipos de sensores y dispositivos entre sí, para una comunicación compleja e integral de sistemas con diferentes tipos de tecnologías (Ferrigno et al., 2011; Vujovic & Maksimovic, 2014). WSN fue un paso firme para el desarrollo de aplicaciones y tecnologías autónomas como lo son los robots de enjambre, esta tecnología consiste en la comunicación e interacción inteligente de flotillas de robots que estén realizando actividades específicas o en conjunto en una misma área geográfica con la meta de que se generen patrones de comportamiento colectivo mediante las interacciones mutuas entre robots y de robots con su entorno sin obstruirse a sí mismos o en sus actividades (Le & Ngo, 2017; Patil et al., 2015).

Actualmente CETYS Universidad campus Mexicali se está trabajando en un proyecto de investigación que tiene por objetivo el diseño y desarrollo de un vehículo autónomo. El objetivo de este proyecto es dar un nuevo servicio de transporte autónomo a los miembros de la comunidad o para el transporte de material a través del campus universitario. En este proyecto se han utilizado tecnologías accesibles para la automatización de sistemas que puedan ayudar a cumplir con las metas planteadas del proyecto el cual es el desarrollo de un vehículo autónomo de una manera económica y eficiente (Basaca-Preciado et al., 2018). Para ello se han utilizado diferentes dispositivos que realizan distintas tareas específicas que se requieren para el funcionamiento

correcto del vehículo. El proyecto actualmente cuenta con dos versiones prototipos del vehículo, los cuales fueron un gran inicio para este proyecto, pero con las últimas incorporaciones de hardware se ha podido dar un avance en la autonomía del mismo. Para ello se requiere mantener constantemente intercomunicados todos los dispositivos, sensores y actuadores del vehículo, además de poder visualizar y guardar información de manera remota. Sin embargo, con las actuales implementaciones de los nuevos dispositivos se ha convertido en una tarea complicada, debido a que ha incrementado la complejidad del sistema para la recolección de datos e información de los distintos componentes del vehículo, ya que algunos son especializados para ciertas tareas o manejan pocos o un protocolo de comunicación.

En los últimos años, se han hecho desarrollado de módulos adaptadores de las tecnologías WSN a IoT, debido que son utilizadas en las investigaciones y en algunas industrias, pero no se han desarrollado para protocolos de comunicación alámbricas (Kruger et al., 2015; Qingping Chi et al., 2014).

## **1.2 Justificación.**

Los módulos adaptadores son necesario en la actualidad, para conectar los dispositivos y sistemas completos sin módulos de comunicación inalámbrica, a los nuevos paradigmas de las tecnologías de IoT e I4.0, será necesario el uso de un adaptador compatible para este tipo de tecnologías para que puedan entrar en los ecosistemas que se están generando a base de esos paradigmas. Como se mencionó anteriormente se han desarrollado dispositivos adaptadores de tecnologías para Internet de las cosas, pero en su mayoría han sido en tecnologías inalámbricas ya establecidas, pero ninguno como adaptador por protocolos alámbricos, los cuales son los protocolos comunes utilizados dentro de la industria.

En el proyecto del vehículo autónomo se encuentran componentes que tienen módulos de comunicación inalámbrica pero no todos tienen algún módulo, además que todavía en etapa del proyecto no están intercomunicados entre sí, lo cual dificulta el monitoreo de información y señales del vehículo. Para ello con el trabajo de esta tesis se podrá obtener un módulo adaptador que pueda integrar los dispositivos a un ecosistema

de IoT, además de una segunda función de comunicar todos los componentes entre sí, para tener una mejor comunicación dentro del vehículo.

Las ventajas que puede brindar el uso del enfoque de Internet de las cosas a este proyecto es conveniente, debido que se puede facilitar diferentes tareas de procesamiento liberando gasto computacional y economizando en uso de hardware especializado, como ejemplo de estas tareas puede ser el subir información a la nube y procesar en ella misma la información para la caracterización de comportamientos de los sistemas o componentes, despliegue de información en algún tablero interactivo, enviar alertas por cualquier inconveniente identificado, o inclusive poder mandar comandos de control al vehículo. Para la integración de todos los componentes del vehículo se requiere un módulo adaptador que pueda interactuar con ellos, además de interactuar con distintos protocolos de comunicación alámbricos debido que la mayoría no tienen módulos de comunicación inalámbrica.

### **1.3 Planteamiento del Problema.**

Actualmente se pueden encontrar dispositivos adaptadores a IoT, con propósitos generales como específicos, también tanto comerciales como de investigación. La mayoría de los dispositivos adaptadores de investigación se enfocan en los protocolos de comunicación inalámbrica ya que el propósito de estos dispositivos es adaptar las WSN a IoT, por otro lado, son muy pocos los protocolos de comunicación inalámbricos integrados a estos adaptadores. Los comerciales se enfocan más a las necesidades de la industria, por lo cual sus especificaciones varían dependiendo el área de enfoque aunque en algunas se pueden encontrar protocolos alámbricos de comunicación, como compatibilidad con microcontroladores, pero a un alto precio.

En el proyecto del vehículo autónomo de CETYS Universidad, está en proceso de su tercera versión la cual ya tendrá un modo de transporte autónomo, para ese fin, se está integrando diversos equipos de óptica, geolocalización, sensores ópticos, entre otros. Los componentes incorporados últimamente se han probado de manera individual y aislada del sistema principal, y no todos los componentes tienen un módulo de

comunicación inalámbrica, por lo que dificulta la operación interna del vehículo debido a que se requiere la comunicación entre dispositivos.

En prototipos la primera versión del vehículo se diseñó para operar de manera manual, para lo cual no requería la intercomunicación entre todos los dispositivos para hacer operable, en la segunda versión, se agregaron más componentes como sensores, microcontroladores, para poder enviar información desde los sensores a los motores pasando por el módulo de control, haciendo posible el control por tiempo y distancias. Además, se necesita un sistema de monitoreo remoto y de adquisición de datos, ya que en pruebas iniciales se ocupa de una computadora arriba del vehículo, lo cual era un factor importante para la duración de las pruebas.

Es por tal motivo que este proyecto de tesis plantea el desarrollo de un módulo adaptador compatible con los componentes comerciales del vehículo y sus protocolos, además de contar con un módulo de comunicación de Wi-Fi para su uso como Gateway para IoT. Al incorporar estos cambios se obtendrán mejoras sustanciales en el vehículo, por ejemplo, la eliminación del ordenador a bordo y dejar solamente el módulo de control, y desplegar la información del vehículo remotamente por medio de IoT. Con esto como base, puede tener usos en diferentes tareas, como robots de enjambre, robot estructural (BIM), puntos de lecturas remotos, entre otros.

#### **1.4 Preguntas de investigación.**

¿Cuántos dispositivos serán compatibles con el dispositivo adaptador?

¿Qué protocolos serán los que se integrarán al proyecto?

¿Qué plataformas en la nube serán compatibles con el dispositivo?

¿Qué se podrá interactuar de manera remota al vehículo?

## **1.5 Objetivos.**

### **1.5.1 Objetivo General.**

Desarrollar un módulo adaptador para poder enviar información a una plataforma digital donde se pueda realizar un monitoreo e interacción de datos y comandos a larga distancia con múltiples dispositivos, sensores y actuadores, trabajando en la integración de diferentes protocolos de comunicación por medio de servicios de nube e Internet de las cosas.

### **1.5.2 Objetivos Específicos.**

- Comparar desempeños de protocolos de comunicación con interfaz alámbrica (I2C, SPI, UART, CAN), para su selección para comunicar entre sí todos los dispositivos e implementación en el vehículo autónomo de CETYS Universidad.
- Generar una metodología para la adaptación de dispositivos con protocolos alámbricos a hacia IoT.
- Integrar las nubes comerciales como código abierto (open source) para el almacenamiento y despliegue de información en un tablero.

## **1.6 Hipótesis.**

La implementación de un sistema embebido como dispositivo adaptador de Internet de las cosas, permitirá la reducción de componentes internos de un vehículo autónomo, la optimización de los recursos y funciones de cada componente mediante la simplificación de la arquitectura de comunicación, el ahorro en consumo de energía y la adaptación de bajo costo a ecosistemas IoT.



## **Capítulo 2. Marco Teórico.**

### **2.1 Internet de las cosas.**

Internet de las cosas (IoT), es un paradigma tecnológico que representa la intercomunicación de dispositivos de uso cotidiano por medio redes inalámbricas e Internet, con el objetivo de recabar información de lo que sucede a su alrededor, creando un ambiente invisible de múltiples conexiones donde trabajan dispositivos, máquinas, sensores, y demás equipo electrónico. En el 2009 se tenían 900 millones de dispositivos conectados a Internet y se tienen diversos pronósticos de cuantos dispositivos se tendrán en este año 2020, las predicciones apuntan a 50 mil millones de dispositivos, otros entre 26 mil millones y 24 mil millones (Gubbi et al., 2013; Lee & Lee, 2015; Nordrum, 2016).

IoT ha crecido en gran escala en los últimos años, evolucionando a un paso increíblemente rápido. A continuación se presentan 5 tecnologías que han favorecido para el crecimiento de IoT: Identificación por radio frecuencia (RFID), Red de sensores inalámbricos (WSN), Middleware, Cómputo en la nube, y Software de aplicación IoT (Lee & Lee, 2015).

#### **2.1.1 Identificación por radio frecuencia (RFID).**

La Identificación por radio frecuencia permite la identificación automática y captura de datos por medio de ondas de radio, con tags y lectores. RFID frecuentemente se utilizan tags y lectores para la captura de las ondas de radio, como se demuestra en la fig. 1, los tags pueden ser pasivos o activos, los pasivos no tienen una fuente de poder propia entonces los lectores deben energizar estos tags pasivos. Los tags activos tienen una fuente de poder integrada lo cual les permite operar en frecuencias más altas que las pasivas, además de tener más distancia de rango hasta los cien metros. Se pueden encontrar en cualquier lugar RFID tags, ya sea para la industria, para localización de materiales, para asignación de id a artículos. También podemos encontrar en objetos cotidianos, como en las llaves de los vehículos, en las tarjetas de transporte, como camiones, metros, trenes, en id de empleados (Weinstein, 2005).

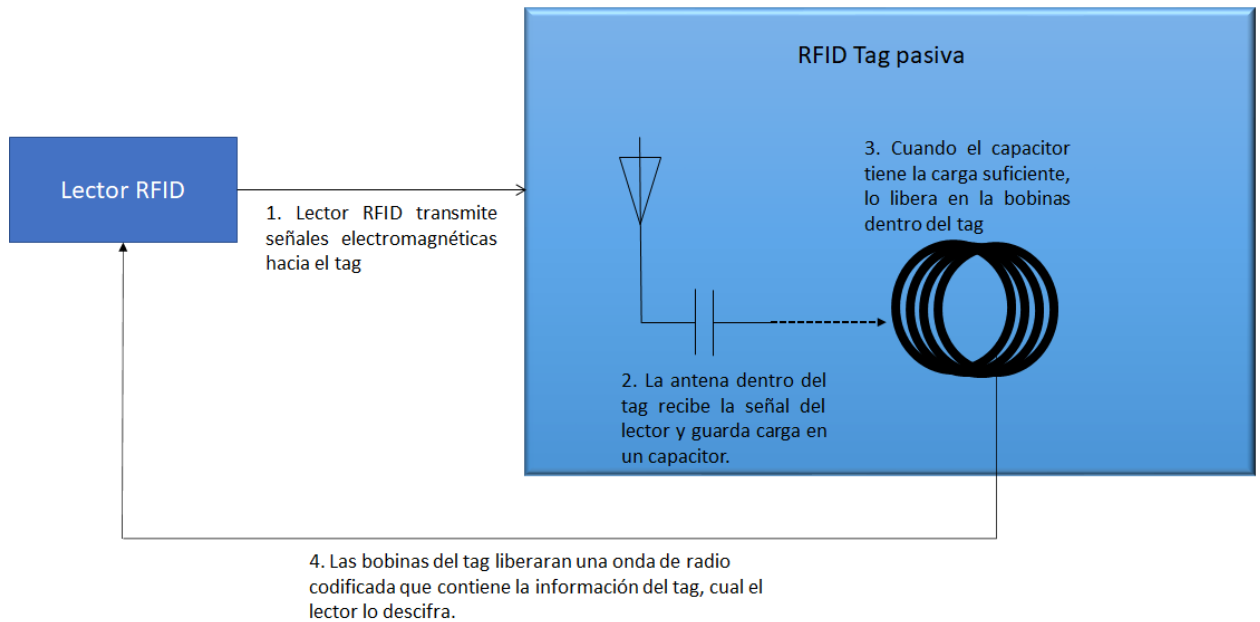


Fig. 1. Vista simplificada de transferencia de datos en una RFID tag pasiva de baja frecuencia.

Fuente: Elaboración propia.

### 2.1.2 Red de sensores inalámbricos (WSN).

La red de sensores inalámbricos consiste en dispositivos equipados con sensores autónomos distribuidos espacialmente para monitorear las condiciones físicas del ambiente y que pueda colaborar con los sistemas de RFID para tener un mejor seguimiento de los estados de las cosas, como localización, movimientos, temperaturas, entre otros. En los últimos años, se han desarrollado estándares para adecuar de mejor manera las WSN a los pisos de manufactura de las empresas, de las cuales resaltan dos WIRELESSHART y ISA100.11A. Las diferencias son WIRELESSHART todos los dispositivos deben tener un router, como se puede observar en la figura 2a, en cambio ISA100.11A, los dispositivos no pueden contar con un router, por lo tanto sus redes son distintas al momento de utilizar en un ambiente real, en donde WIRELESSHART será una única red, mientras ISA100.11A, tendrá subredes que conforman una gran red abarcando los dispositivos sin conexión, en la fig. 2b se puede ver esta cobertura de red (Adriano et al., 2018).

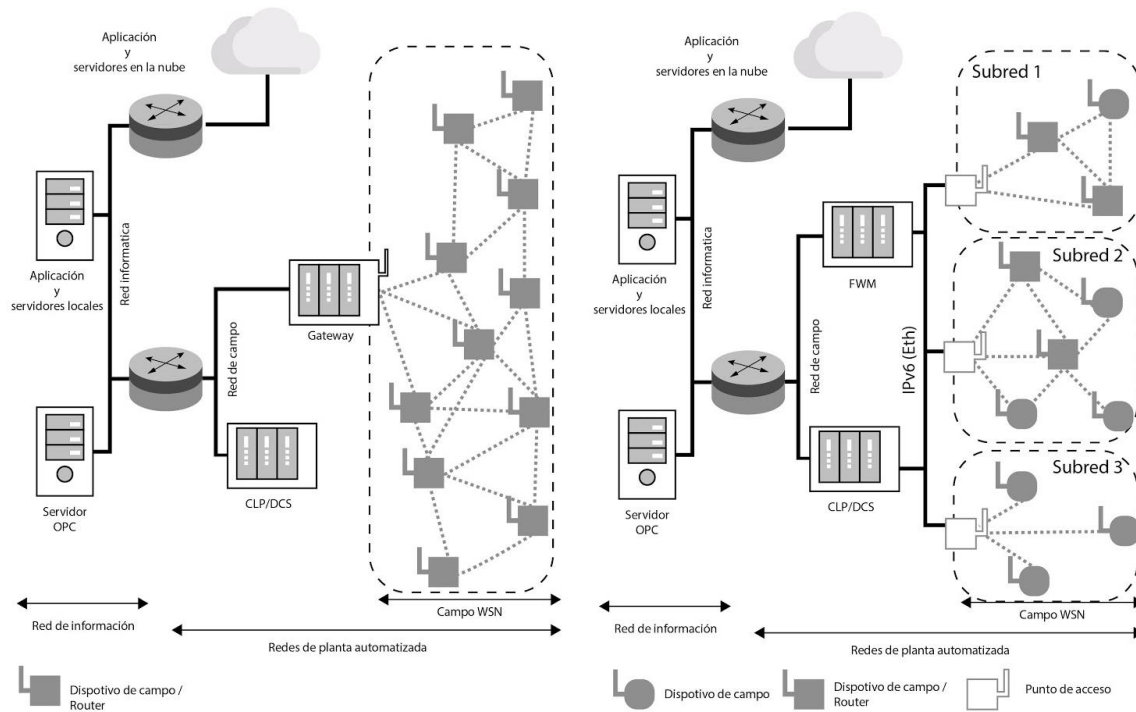


Fig. 2.a Arquitectura de red WIRELESSHEART para aplicación de Industria 4.0.

Fig. 2.b Arquitectura de red ISA100 para aplicación de Industria 4.0.

Fuente: Elaboración propia.

### 2.1.3 Middleware.

Middleware es la capa de software que se interpone entre las aplicaciones de software para facilitar la comunicación de entradas y salidas a los desarrolladores de software, en la fig. 3 se puede ver un ejemplo dentro de la industria con una grúa parcialmente autónoma. Los middlewares se pueden encontrar en una gran variedad de tipos de software, por ejemplo, para interactuar con bases de datos, mensajería, sitios webs, entre otros. Como requisitos deben ejecutar los comandos en un tiempo apropiado sin un retraso notable, además de poder administrar y controlar múltiples servicios, otro punto, deberá de facilitar la comunicación entre los servicios tanto internos como externos, puede ofrecer una interfaz uniforme para las aplicaciones de servicios, por último, deberá poder comunicarse con otros middlewares a través de la red del sistema (Glock et al., 2019).

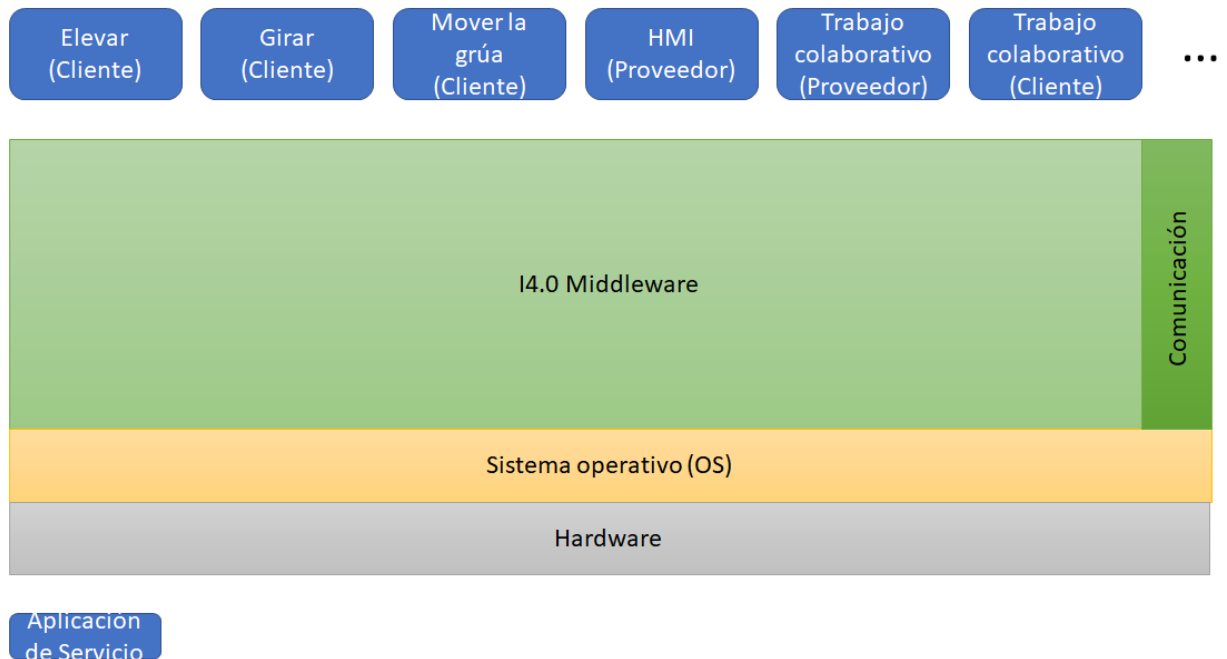


Fig. 3. Arquitectura de software de una plataforma de servicio de Industria 4.0.  
Fuente: Elaboración propia.

### 2.1.4 Cómputo en la nube.

Cómputo en la nube es un modelo para acceso en demanda para compartir recursos configurables, como almacenamiento, procesamiento, servidores, aplicaciones, entre otros, debido que muchas aplicaciones de IoT requieren de almacenamiento masivos para la información capturada, procesamientos en tiempo real, y otros procesos que este modelo puede ofrecer. Este tema tuvo sus comienzos desde los años 50, se le atribuye a John McCarthy, el cual utilizaba el término “Time Sharing”. En los años 60 se popularizó el proporcionar recursos compartidos, donde se alquilaba el tiempo y servicio de cómputo, a los usuarios se les cobraba una cuota por la máquina de cómputo, horas de conexión y el uso de almacenamiento (East Coast Polytechnic Institute, 2020). No fue hasta la década de los 90 por parte de las empresas de telecomunicaciones que pudieron extender el cómputo en la nube hasta cubrir servidores, así como la infraestructura de la red. En los 2000 fue cuando Amazon Web Services, Google, trajeron el cómputo en la nube como lo que es hoy en día (FayerWayer, 2019).

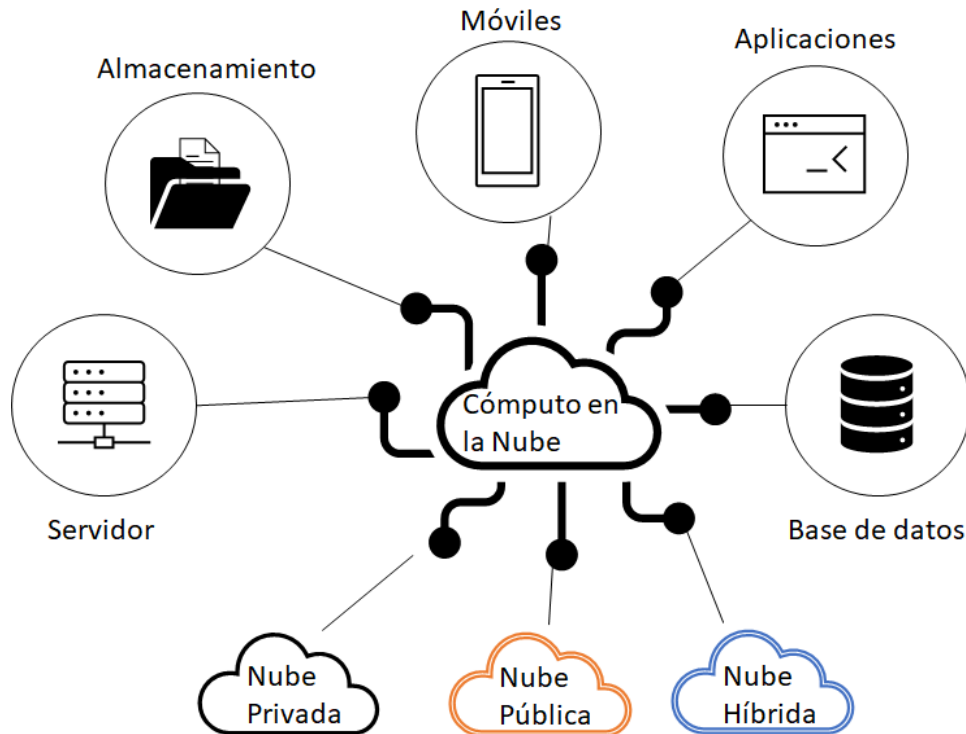


Fig. 4. Aplicaciones y funciones de cómputo en la nube.  
Fuente: Elaboración propia.

El cómputo en la nube puede representarse como en la fig 4, donde se puede ver sus principales usos y aplicaciones además de sus conexiones con todo tipo de nube.

### 2.1.5 Software de aplicación IoT.

Por último, software de aplicación de IoT en dispositivos necesitan asegurar que la información o mensajes sean recibidos y realizados de una manera correcta en el tiempo pronosticado, para facilitar el desarrollo de aplicaciones de IoT para específicas a usuarios u orientadas a la industria. No solamente, las aplicaciones se deben interactuar en los sistemas dispositivos a usuario, sino que también puede utilizarse en sistemas dispositivo a dispositivo, o sistema. Para ello es importante que las aplicaciones IoT deben ser construidas con inteligencia para que los dispositivos puedan monitorear el ambiente, identificar los problemas, comunicarse unos con otros (dispositivos, sistemas o programas), resolver problemas sin la necesidad de una interacción humana (Lee & Lee, 2015).

En los años 80s en la universidad de Carnegie Melon en Pittsburgh, Pensilvania, un grupo de estudiantes desarrollaron un sistema para conectar una máquina expendedora de sodas a APRANET (de las siglas Advanced Research Projects Agency Network), entonces se le considera a este sistema como el primer de IoT. En 1999, fue cuando se utilizó el término “IoT” en una presentación de Kevin Ashton hecha para Procter & Gamble (Goddard, 2019).

Con el uso de la informática en la vida cotidiana y equipos especiales para las generaciones de redes locales, surgieron diferentes propuestas iniciales para ampliar estas redes locales a redes que pudieron interceptar grandes cantidades dispositivos, para ello, se tenían las redes de sensores, integraciones de bases de datos relacional, plataformas basadas en eventos, algoritmos de proximidad y/o localización, tags de RFID, sensores, NFC y sensores en dispositivos móviles, procesamiento de datos serial y paralelo. Posteriormente, se desarrollaron redes autoorganizadas y autoconscientes, con retrasos tolerables de almacenamiento y suministro, algoritmos componibles, módulos de programación a gran escala, programas que pueden ser lectores de múltiples protocolos y múltiples estándares, procesamiento de datos con contexto adaptable. Por último, se han desarrollado redes conscientes del contexto, programación orientada al objetivo, inteligencia distribuida, resolución de problemas, en la parte de sensores se desarrollaron sensores inteligentes, más pequeños e igualmente que los actuadores. Se tiene pronosticado, para después del 2020 que existan redes cognitivas, autorreparables, aprendizaje no supervisado, programación orientada al usuario, IoT invisible, nanotecnología y optimización de procesos cognitivos de datos (Lee & Lee, 2015).

Actualmente muchas empresas han aprovechado las propiedades y ventajas que ofrece Internet de las cosas para la industria y el comercio. Algunos casos de ejemplos serían, el primero en la industria sería la empresa ABB para el monitoreo de robots, así como sus mantenimientos preventivos y cambios de piezas antes que se rompan. Otro caso es North Star BlueScope Steel que hace equipos de seguridad, para prevenir accidentes, descubrir zonas de peligro y tener un seguimiento de la salud de las personas (Buntz, 2017).

## **2.2 Aplicaciones IoT áreas Industrial y Comercial.**

A continuación, se explicará un poco más a detalle aplicaciones de IoT en el área de la industria a la vez de qué es la Industria 4.0, además, se hablarán de las aplicaciones que aprovechan de Internet de las cosas en el área comercial y, por último, se describe el proyecto del vehículo autónomo que se encuentra en desarrollo en CETYS Universidad.

### **2.2.1 Industria 4.0.**

La industria ha tenido 3 revoluciones previas a través de los años, donde los procesos de productividad industrial han tenido grandes avances que marcan un antes y un después de cada revolución. En la primera revolución se caracterizó por el uso de la energía hidroeléctrica, para el desarrollo de máquinas de vapor que trabajaran como herramientas. El sector textil fue el que tuvo más impacto en su desarrollo de producción con la primera revolución industrial. En la segunda revolución industrial fue definida por el uso de la electricidad, la producción en masa y líneas de montajes, además de comenzar con la producción de productos químicos. Tecnologías como el telégrafo, vías de tren crecieron a gran escala durante esta etapa por lo cual ayudó al avance de las ideas y de las personas, terminando en una forma de globalización, otro sector que creció bastante fue el automovilismo y las aportaciones de Henry Ford y su modelo T100, que gracias a las herramientas desarrolladas en esta etapa ayudó a la producción en masa. La tercera revolución industrial fue la informática, con la cual trajo la automatización, el uso de tecnologías de la información y la comunicación, y el uso de dispositivos electrónicos. Dio inicio al desarrollo de herramientas para la fabricación como para el diseño, por ejemplo, máquinas de Control Numérico por Computadora (CNC), Control Lógico Programable (PLC), sistemas de diseños asistidos por computadora, programas de simulación de producción, de calidad, etc. (Zhou et al., 2015).

La Industria 4.0 (I4.0) marca las tendencias de desarrollo del futuro dentro de la industria para lograr más procesos de fabricación inteligentes, incluyendo los sistemas ciber físicos (CPS), y a la construcción de sistemas de producción ciber físicos (CPPS), y la operación e implementación de fábricas inteligentes. La Industria 4.0 es un sistema

complejo y flexible que envuelve una gran diversidad de tecnologías como fabricación digital, redes de comunicación, automatización y muchas otras áreas. Las bases de la implementación de esta revolución están en el diseño digital y simulación, procesos de fabricación altamente autónomos, manejo de datos de producción y administración de procesos de producción (Zhou et al., 2015).

A continuación, se mencionan tecnologías claves para la I4.0; sistemas de producción ciber físicos (CPPS), el Internet móvil y el Internet de las cosas, tecnologías de cómputo en la nube, y Big Data y técnicas de análisis avanzadas. Un primer paso es emigrar la tecnología debido que gran parte de la industria se encuentra con el uso sistemas ciber físicos (CPS) y es necesario para construir una plataforma CPPS, que tiene como objetivo el conectar el espacio virtual con el mundo físico, habilitando la maquinaria y equipo dentro de una fábrica inteligente para ser aún más eficiente, para crear mejores condiciones de producción encaminado a una producción inteligente. La siguiente tecnología es el Internet móvil e IoT, debido a que desde hace tiempo se ha influenciado los sistemas de la industria moderna, a la interacción entre humanos y máquinas para la fabricación inteligentes por medio de aplicaciones móviles, y terminales móviles, donde un operador puede utilizar a distancia, almacenar grandes cantidades de datos automáticamente, con las tecnologías móviles como 3G, 4G y la nueva 5G, con cada nueva generación se tiene más capacidades para transferir información, más tipos de datos compatibles, mayor cantidad de ancho de banda, mayores velocidades de transferencias, además de tener mejores coberturas, todo esto ayudando a aplicaciones con dispositivos remotos sin accesibilidad a una conexión a una red fija y aprovechando las ventajas de las redes móviles.

La tercera tecnología es la de cómputo en la nube, esta tecnología es usada como plataforma digital para ofrecer un rendimiento flexible a la demanda, y el compartir recursos a distintas instancias como sea requerido. Algunas empresas como Amazon Web Services, Google Cloud, entre otras ofrecen servicios en la nube, como almacenamiento, procesamiento, servidores en la nube que se adaptan a las necesidades del cliente, por ejemplo, en horas pico de trabajo puedan tener más recursos para satisfacer la demanda de cómputo. Por último, se encuentra Big Data y las



técnicas de análisis avanzadas, en los procesos de fabricación son instalados uno o más microprocesadores en múltiples máquinas para la recolección de información de producción, con lo cual se generan grandes cantidades de información que ya no se puede procesar sin el uso de la tecnología de Big Data. La Big Data consiste en obtener información de varios tipos de bases de datos en tiempo real, y con ello tomar mejores decisiones, que puedan traer beneficios a las compañías, como procesos optimizados, reducción de costos, entre otros (Zhou et al., 2015).

Una arquitectura para una fábrica inteligente (se le asocia smart factory a una fábrica con tecnologías de I4.0 ya implementadas), se basa en 3 capas; la capa de recursos físicos, capa de red y la capa de aplicación de datos. La primera capa incluye todos los recursos de fabricación involucrados en todo el ciclo de vida de fabricación, que representa la base para el logro de la fabricación inteligente. La siguiente capa es la caracterización por percepción y control, que juega un rol importante en la fabricación inteligente, ya que en la industria se integran una gran variedad de tipos de tecnologías de redes. En la última capa, es el descubrimiento de conocimiento a base de los datos de los recursos y construir una cadena de valor industrial (Chen et al., 2018).

Con los cambios que se está presentado en la industria con su cuarta revolución, se han estado acoplando los procesos a las herramientas que ofrece Internet de las cosas, para mejorar en sus productos, sus tiempos de fabricación como con los proveedores, y a sus procesos de mejoras continuas.

### **2.2.2 IoT Aplicaciones Comerciales.**

Internet de las cosas ha revolucionado tanto la industria como el área comercial, debido a que ha transformado en gran medida la manera de hacer compras en línea, los inmobiliarios de las tiendas y la comunicación entre el cliente y el vendedor (Tate, 2018). En el área comercial se separa los productos o clientes de Internet de las cosas con los inmobiliarios, los productos de Internet de las cosas son los artículos, componentes, dispositivos o sistemas que están comunicados por un protocolo inalámbrico para el uso de monitoreo personal, o conectados a un sistema inteligente como una casa o empresa (Tate, 2018).

En el área de inmobiliarios, también se refiere a la manera que opera y ofrece sus servicios la empresa comercial. Por parte del servicio, se orienta a una interacción personal con cada consumidor, desde la asistencia en la compra, como recomendaciones de productos, seguimientos de envíos, soporte posterior a la venta. La inteligencia artificial es un gran factor en esta parte de los servicios debido que con los visitas y búsquedas que haga el consumidor se puede ofrecer mejores productos a sus necesidades y gustos (Tate, 2018), en la fig. 5 se puede ver las actividades principales que se han desarrollado con el IoT en el comercio. Ahora en si los inmobiliarios, se han transformado debido que no es necesario tener un local físico abierto al público. Las empresas, están usando oficinas inteligentes las cuales tienen sensores y actuadores conectados a la nube, desde esas oficinas ofrecen sus servicios y soporte a los clientes (Knight, 2015), dentro de la fig. 6 se muestran las aplicaciones a través de IoT en los edificios inteligentes. Además, tener procesamiento e información en nubes privadas el cual es un recurso útil para recopilación y manejo de datos, ya que todo el procesamiento para ofrecer productos o dar soporte a un cliente está en la nube y no debe estar todo en un solo punto físico, haciendo flexible el análisis y monitoreo de datos, tomas de decisiones, y por último, hacer el comercio escalable de una manera sencilla (Tate, 2018).

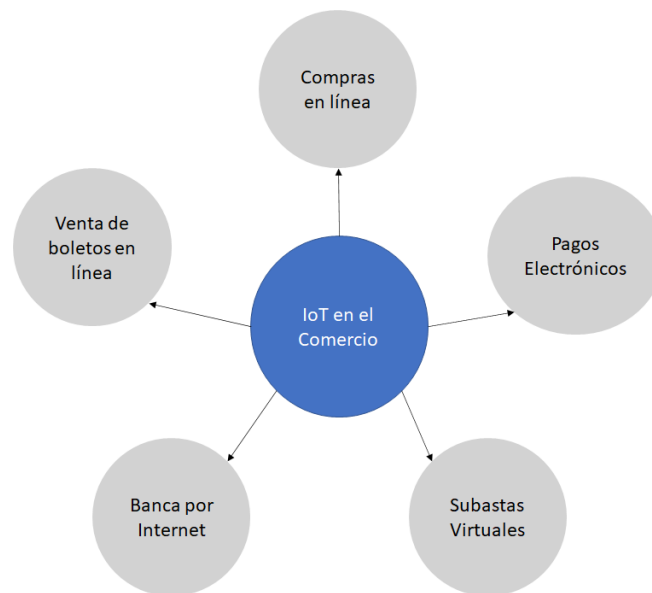


Fig. 5. Actividades de IoT en el Comercio.  
Fuente: Elaboración propia.



Fig. 6. Aplicaciones de IoT en edificios inteligentes.  
Fuente: Elaboración propia.

Los productos de Internet de las cosas para el uso de consumidores resaltan empresas como Google, Amazon, Apple, Fitbit, entre otros. La mayoría de los productos que se encuentran en esta categoría son wearables para el seguimiento o monitoreo de la persona, y están conectados a un dispositivo inteligente o a una nube directamente o ambos (Fitbit, 2020). Otro producto muy presente son las inteligencias artificiales, que ofrecen varias empresas reconocidas en el mercado actual, las cuales pueden ayudar desde la agenda de la persona, como pedir ayuda en situaciones de emergencias (Apple Inc, 2020). Otros productos del mercado actual es el uso de equipo de limpieza autónomas y con control remoto, timbres de entradas que cuentan con cámaras para el enviar imágenes y videos a la persona de lo que pasa enfrente de su casa, termostatos inteligentes y con control por internet (Nest, 2020). La mayoría de estos productos pueden ser usados e interactúan con el usuario por medio de su teléfono inteligente aprovechando las redes móviles y redes de internet para un mejor acoplamiento de uso con el usuario.

## 5 beneficios de Internet de las cosas en el comercio (Matthews, 2019):

- Edificios más eficientes.
  - La optimización y automatización de la funcionalidad del edificio, como los sistemas de aires, de iluminación, de seguridad, es uno de los primeros y más atractivos pasos que una empresa puede aprovechar de Internet de las cosas.
- Brindar Inteligencia a la cadena de suministro y gestión de activos.
  - Con las tecnologías RFID y GPS se obtienen más datos sobre la producción de productos, seguimientos de vehículos de transporte y de productos.
- Mejoramiento en la percepción y la conversión con el cliente.
  - Igualmente, con la ayuda de las tecnologías de IoT, se puede ofrecer servicios, productos o descuentos a los clientes por su posición geográfica, o sus búsquedas en Internet.
- Facilidad en la toma de decisión en tiempo real e incremento de seguridad en conformidad.
  - En la cadena de suministro, envíos de productos, en las retroalimentaciones de los clientes sobre los productos o servicios se pueden realizar decisiones en tiempo real para el mejoramiento del servicio, así como el aumento de la conformidad con el producto.
- Mejoramiento en la seguridad en el lugar de trabajo y control de acceso.
  - Con la ayuda de los dispositivos de seguridad y sensores de alarmas conectados a la nube ofrecen una mayor seguridad y tiempo de respuesta en casos de emergencias.

### **2.2.3 Vehículo autónomo CETYS Universidad.**

En el campus se está desarrollando un proyecto de investigación, el cual consiste en el desarrollo e implementación de vehículos autónomos para el transporte de la comunidad académica dentro del mismo campus. El proyecto se encuentra en su tercera etapa, en el cual se están integrando diversos equipos especializados para la autonomía del vehículo, como LiDAR, GPS especializados, cámaras, sensores. En etapas anteriores se había alcanzado hasta un nivel 2 de autonomía, la primera etapa fue controlado de manera manual por un usuario arriba del vehículo, en la segunda etapa se implementó un control de tiempo con la lectura de las llantas para tener un módulo de control para el giro de las llantas y avanzará de manera constante.

El vehículo autónomo tiene como objetivo ser un sistema de transporte inteligente implementado en una estructura automóvil capaz de transportar personas desde un punto A hasta un punto B. El diseño de una infraestructura de red que sea suficiente para todo esquema de comunicación y cumplir con los requisitos para un campus inteligente. Por último, construir un sistema eléctrico sostenible para el vehículo, con una batería capaz de recargarse a través de energía solar, con las que contará el vehículo y con las que se han estado instalando en el campus (Basaca-Preciado et al., 2018).

Con respecto al esquemático de interconexión general del vehículo, se encontrará un módulo de control, el cual en este caso es un roboRIO de National Instruments. Este dispositivo tendrá conectado una gran variedad de sensores y actuadores, como encoders, sensores de proximidad, controladores de velocidad de los motores, GPS, cámaras, entre otros. Anteriormente, el proyecto de investigación había planteado incorporar un cómputo u ordenador interno de alta capacidad de procesamiento dentro del vehículo, pero se ha descartado principalmente por el consumo energético y la batería que limita su uso, además de reducir el costo del vehículo, por lo tanto, la comunicación con el servidor se ha visto afectado. Por este motivo el desarrollo de este proyecto de un módulo adaptador para poder adaptar la mayoría de los dispositivos que trabajan internamente en el vehículo autónomo, como adquisición de datos e intermediarios, y al mismo tiempo su controlador reconfigurable de puertos integrados, debido que no cuenta con un módulo Wi-Fi.

## **2.3 Dispositivos Adaptadores.**

En los últimos años se han desarrollado una gran diversidad de dispositivos adaptadores para Internet de las cosas, estos tienen desde enfoques comerciales hasta su uso exclusivo en el área investigación científica. En esta subsección se mencionan algunos dispositivos adaptadores que se han desarrollado en el área de investigación como comerciales, para tener un mejor conocimiento de las opciones actuales.

### **2.3.1 Comerciales.**

Con el gran impacto de la Industria 4.0, además de la construcción de casas y edificios inteligentes se han desarrollado distintos dispositivos adaptadores para cada área de enfoque. Desde el 2017 varias empresas han surgido para el desarrollo de estos dispositivos, como EUROTECH, SIERRA WIRELESS, B-Scada, SMARTWORX. Grandes empresas del área de las tecnologías han creado departamentos para el desarrollo de estos dispositivos. El enfoque principal hasta ahora de los dispositivos adaptadores comerciales es hacer puentes entre la maquinaria, equipo, y dispositivos ya en un ecosistema, para conectarlos a Internet o a una nube digital, comunicándose con los sistemas telemáticos, protocolos específicos, y algunos protocolos inalámbricos, como, Bluetooth, ZigBee, GPS (i-SCOOP, 2020).

En el mercado se pueden encontrar empresas que se especializan en el servicio de software y análisis, como empresas dedicadas en hardware, y algunas que ofrecen productos y servicios de principio a fin. Algunas empresas reconocidas como DELL, HUAWEI, INTEL, HP Enterprise, son algunas que han desarrollado dispositivos adaptadores, como software para el mercado (Harwood, 2019).

La empresa Lanner ofrece hardware para dispositivos adaptadores con microprocesadores de Intel, que tienen conexiones de puertos de LAN, USB, SERIAL, 4G/LTE, y POE. Estos componentes son sistemas embebidos x86, los cuales pueden operar con cable LAN y operar inalámbricamente con Wi-Fi y 4G/LTE. Los usos que se le pueden dar son implementación para aplicación IoT como señalización digital

inteligente, análisis de ventas de tiendas, automatización de edificios y control de acceso en condiciones comerciales (LEI Technology, 2020).

Eurotech es una empresa proveedora de principio a fin en adaptadores para IoT. Es un proveedor de sistemas embebidos, soluciones para Internet de las cosas y plataformas de máquina a máquina (Hegde, 2017). Esta empresa tiene una línea de productos con microcontroladores reprogramables, los cuales pueden programarse en JAVA, las comunicaciones y protocolos que se pueden utilizar varían dependiendo del producto, así como los consumos de energías de cada uno, pero se pueden encontrar con protocolos seriales, CAN, Bluetooth, LTE, Wi-Fi, puertos LAN, USB, y en algunos productos tienen entradas y salidas digitales (EUROTECH, 2020).

NXP es una empresa que ofrece soluciones para conexiones de seguridad y sistemas embebidos, y uno de su hardware es un dispositivo adaptador, el LS1021A-IoT Gateway, el cual tiene puertos LAN, USB, serial, compatibilidad con carcasas de Arduino, SPI, I2C, entradas digitales y analógicas, además de entradas y salidas de propósito general (GPIO). Utiliza un microcontrolador para su programación, no tiene un módulo de Wi-Fi, pero se puede agregar un módulo por medio de PCIe, pero si maneja el protocolo ZigBee, y también tiene salida HDMI para desplegar cualquier información. Este es el de los dos productos que se mencionan, que puede ser comprado de manera directa sin solicitar un presupuesto previamente, pero el precio base es de 595.16 dólares americanos por este kit (NXP Semiconductors, 2020).

La empresa seeed también ofrece un kit de desarrollo para un dispositivo adaptador para IoT, el cual también se basa en un procesador Intel para el microcontrolador, que es compatible con Bluetooth, Wi-Fi, PCIe, USB, LAN. Este kit maneja un sistema de Plug-N-Play para los sensores y actuadores que se pueden utilizar. Por otra parte, tiene soporte para la nube de Microsoft Azure y sus servicios IoT que esta ofrece, el precio del kit es de 300 dólares americanos, pero ya tiene algún tiempo fuera de disponibilidad (Seeed Technology Co., 2020).

### **2.3.2 Desarrollo Investigación.**

En la comunidad científica se han desarrollado múltiples dispositivos adaptadores para funciones específicas para velocidades de transferencias, tipos de datos, o conexiones, como con propósitos generales para abarcar los usos comunes en el área. La mayoría de estos dispositivos se han apoyado de las tecnologías de Wireless Sensor Networks (WSN) para adaptarlas a ecosistemas de Internet de las cosas, debido que las WSNs tienen sus protocolos estandarizados, además de tener ecosistemas completos con una gran diversidad de dispositivos o nodos comunicados inalámbrica. Por otro lado, las tecnologías WSN fueron implementadas en varios campos industriales y de consumo, como monitorización y control de procesos industriales de comunicaciones inalámbricas, por lo cual también ayudó al desarrollo de dispositivos adaptadores para estas tecnologías, aparte que era más sencillo de adaptarlas.

Desde principios de los 2000's ya había interés en crear dispositivos para que se comunicaran a internet, con propósitos de automatizar casas (Alkar & Buhur, 2005), como también crear redes de sensores embebidos (Savvides & Srivastava, 2002).

En esos tiempos todavía no había protocolos estandarizados, por lo tanto, no existían opciones de comunicación inalámbricas, por lo tanto, requerían de una computadora para la conexión a Internet y para conectarse al computador era por el puerto RS-232. Para la comunicación inalámbrica comúnmente se utilizaba radio frecuencia ya que había módulos de bajo consumo energético.

En la década pasada comenzaron a hacer dispositivos adaptadores para las WSN, para aprovechar todas las ventajas y recursos que esta tecnología presenta, además de su infraestructura que ya estaba en muchos sistemas. Para ello comenzaron con sistemas embebidos para cubrir las necesidades, algunos con microcontroladores complejos para usarlo como controlador principal (Qingping Chi et al., 2014), para estos dispositivos los puntos clave son la lectura en paralelo y en tiempo real con múltiples velocidades para la lecturas de sensores. También hubo otros que comenzaron a aprovechar los microcontroladores que se encontraban en el mercado e utilizando adaptadores de comunicación inalámbrica como ZigBee, más módulos de Wi-Fi, y



programando el microcontrolador a lenguajes como Python para poder obtener un dispositivo adaptador (Wibowo et al., 2014).

Otro dispositivo de desarrollo fue una microcomputadora para su uso como adaptador es el Raspberry Pi. Este se puede encontrar en una gran diversidad de artículos de prototipos funcionales con esta microcomputadora. Una ventaja de esta microcomputadora es que tiene módulos de Wi-Fi y Bluetooth ya integrados, por lo cual ya tiene un protocolo de comunicación muy usado en las WSNs. Lo interesante de cada uno de los prototipos es su manera de comunicarse con la nube, pero todos son diferentes al momento de comunicarse con los otros dispositivos. Algunos utilizan protocolos de comunicación inalámbrica de muy bajo consumo energético el cual para sitios remotos y que trabajen con una batería es muy beneficioso (Kruger et al., 2015). En otros casos utilizaron la Raspberry como microcomputadora para crear en ella un servidor y además que adquiera los datos de los nodos de la WSNs, de manera económica, aunque esto tiene sus límites ya que para hacer operaciones o toma de decisiones en tiempo real carece de recurso computacional, pero para tareas sencillas o sin mucha complejidad es una gran herramienta (Vujovic & Maksimovic, 2014). Otro ejemplo es usarlo como un adaptador semántico como servicio, esto refiriéndose a comunicarse a la nube por medio de protocolos de comunicación de la capa aplicación del modelo OSI, los cuales pueden ser CoAP, XMPP o MQTT, además cuenta con puertos USB y un GPIO pueden conectarse una gran cantidad de sensores o módulos de comunicación RFID, y por Bluetooth gracias a su módulo integrado (Desai et al., 2015).

## **2.4 Dispositivos / Protocolos / Capa de Aplicación.**

### **2.4.1 Dispositivos**

A continuación, se describirán características de los dispositivos que se encuentran dentro del vehículo de pruebas, enfatizando que protocolos de comunicación tienen y módulos de comunicación inalámbricas que algunos tienen.

#### *2.4.1.1 National Instruments roboRIO.*

El roboRIO es un controlador de robótica reconfigurable, el cual cuenta con puertos integrados para I2C, SPI, RS232, USB, Ethernet, PWN y relés para conectar sensores y actuadores comúnmente utilizados en robótica. El controlador tiene LEDs, botones, acelerómetro interno y un puerto de electrónica personalizado. Contiene un dual-Core ARM real-time Cortex-A9 y el FPGA Xilinx personalizado a una velocidad de 667 MHz, en la parte de memoria contiene 512 MB de almacenamiento, por la RAM utiliza DDR3 con 256 MB a una velocidad de 533 MHz con ancho de bus de 16 bits. El puerto Ethernet es compatible con el protocolo IEEE 802.3 y hasta una velocidad de 100 Mbps, en puerto USB contiene 2 tipo A generación 2 de alta velocidad, en puertos analógicos con tiene una resolución de 12 bits con 4 canales individuales de entrada y 2 de salidas en un rango de operación de 0 a 5 V, en los puertos digitales son 10 para entradas y salidas, además, de las líneas para los puertos UART, I2C y SPI. En su consumo de energía ocupa un voltaje dentro del rango de 7-16 V, en tiempo libre consume 5W y su máximo consumo son 45W (National Instruments, 2015).

#### *2.4.1.2 ATmega 2560 microcontrolador.*

El ATmega2560 es un microcontrolador basado en 8-bit AVR RISC (de sus siglas en inglés Advanced Virtual RISC). Sus características cuentan con 256 KB IPS de memoria flash, 8 KB SRAM, 4 KB EEPROM, 86 líneas de uso de entradas y salidas, 32 registros, contador de tiempo real, 6 contadores flexibles con modos de comparación, PWM, UART, SPI, I2C, convertidores de 10 bit de análogo a digital, JTAG para depuración. Sus velocidades de reloj del CPU son de 16 MHz, utilizando un voltaje entre 4.5-5.5 V (Microchip Technology Inc, 2020b).

#### *2.4.1.3 ATmega 328 microcontrolador.*

El ATmega328 es un microcontrolador basado en 8-bit AVR RISC que cuenta con 32 KB IPS de memoria flash, 2 KB SRAM, 1 KB EEPROM, 23 líneas de uso de entradas y salidas, 32 registros de propósito general, tres contadores flexibles con modos de comparación, además de contar con interrupciones externas e internas, UART, I2C, SPI,

convertidores de analógicos a digital de 10-bit, PWM, su velocidad de reloj del CPU es de 20 MHz a un voltaje de 5.5V que sería su máximo (Microchip Technology Inc, 2020a).

#### *2.4.1.4 WeMo D1 mini microcontrolador.*

Es una mini tarjeta de Wi-Fi con 4 MB de memoria flash basado en un ESP-8266EX, el cual tiene las siguientes características; 11 líneas para uso de entradas y salidas digitales, de las cuales se comparten para utilizar interrupciones, PWM, I2C, UART, además cuenta con una entrada analógica que acepta una entrada de 3.2 V, conexión por Micro USB, es compatible con MicroPython, el IDE de Arduino y nodemcu, opera en un voltaje de 3.3 V, su máxima velocidad de reloj es de 160 MHz (wemos cc, 2019).

#### *2.4.1.5 Raspberry Pi 3.*

El modelo que se pondrá dentro de las pruebas es el Raspberry Pi 3 B+, el cual es una computadora en una tarjeta individual, por lo tanto, puede soportar sistemas operativos de baja demanda como Linux, para ello ocupa una memoria Micro SD, para su fuente de alimentación ocupa 5V con 2.1 A. De conexiones cuenta con HDMI, puertos USB, Ethernet, Salida de audio y video, además de 40 pines para uso generales, como entradas y salidas, I2C, SPI, UART, PWM, también cuenta con conexiones DPI, CSI y DSI para cámaras y pantallas. Su procesador es un Quad Core a una velocidad de 1.2 GHz de 64 bit, con un 1 GB de RAM, el Ethernet es de 100 Mbps, además de contar con módulos de Wi-Fi y Bluetooth de bajo consumo en la misma tarjeta (RASPBERRY PI FOUNDATION, 2015, 2020).

#### *2.4.1.6 Victor SPX (controladores de motores).*

Los Victor SPX son controladores de motores, son controlados originalmente con señales PWM, pero en la última generación se agregó el soporte para el protocolo CAN, también tiene control de voltaje y control de posición con ciclos cerrados. Además de contar con conexión de servos uno macho y uno hembra para facilitar la conexión. Usa un rango de voltaje de 6-16 V a 60 A, la velocidad de entrada es de 2.9 – 100 ms, a una frecuencia de cambio en los engranes de 15 kHz (VEX Robotics, 2020).

#### *2.4.1.7 Optical Encoders.*

Para la lectura de las llantas se utilizan encoders de la marca US DIGITAL, el modelo E4T, debido tiene una carcasa pequeña para aplicaciones donde el espacio sea un limitante, con rangos de 100 a 1000 ciclos por revolución (CPR) y (400 a 4000 pulsos por revolución (PPR), el cual necesita una alimentación de 5 V a 35 mA. Por último, entrega una retroalimentación cuadratura digital para altos volúmenes (US Digital, 2020).

#### *2.4.1.8 Sensores Ultrasónicos.*

Los sensores ultrasónicos son de la marca MaxBotix con un rango de medición de 20 a 765 cm, el cual tiene una zona muerta de 20 cm, cualquier cosa menor a 20 cm mostrará como resultado 20 cm, a parte tiene una resolución de 1 cm. Opera en un rango de voltaje 3.3 a 5.5 V. Tiene salidas para comunicarse con el microcontrolador, a través de voltaje análogo, RS232, PWM, usa una frecuencia de 42 kHz para medir la distancia para los objetos (MaxBotix Inc, 2017, p. 1).

### **2.4.2 Protocolos**

Lo siguiente en describir son los protocolos que se encuentran en los dispositivos dentro del vehículo de pruebas, los cuales se enfatizará en el número máximo de dispositivos que pueden comunicarse simultáneamente por medio de ese protocolo, además el número de bytes que pueden enviarse por interacción, por otra parte, el tipo de comunicación como maestro-esclavo, en un canal central, o de punto a punto.

#### *2.4.2.1 I2C*

I2C (de las siglas Inter-Integrated Circuit), es un protocolo de comunicación alámbrico con un modo multi-maestros, donde 128 dispositivos pueden ser conectados al mismo tiempo entre sí, esto debido a que el protocolo usa 7 bits de direccionamiento, además puede enviar n número de bytes por interacción. I2C puede ejecutarse en tres modos distintos de velocidades; la estándar (100 kb/s), la rápida (400 kb/s), y la alta velocidad (3.4 Mb/s). una característica principal es que solo es requerido dos líneas de conexión para ejecutar este protocolo (SDA y SCL) (Leens, 2009).

#### *2.4.2.2 SPI*

SPI (de las siglas Serial Peripheral Interface), es un protocolo de comunicación alámbrica con un modo de un único maestro, donde la cantidad de dispositivos se puedan conectar depende directamente de la cantidad de señales Selección de Esclavos (SS, Slave Select) que tenga el dispositivo maestro. En frecuencia no tiene un estándar establecido ni un direccionamiento específico. Este protocolo solamente permite la transferencia de un byte por interacción. Adicionalmente, el protocolo SPI permite la comunicación sencilla con los dispositivos esclavos (Leens, 2009).

#### *2.4.2.3 UART*

UART (Universal Asynchronous Receiver and Transceiver), es un protocolo de comunicación alámbrico de punto a punto, por lo tanto, solo soporta la conexión de 2 dispositivos para comunicarse directamente entre ellos. Solo se necesita dos líneas de señales para comunicarse (TXD y RXD), un transmisor y un receptor, puede enviar n cantidad de bytes por interacción. Para su frecuencia en que trabaja maneja un gran rango de frecuencias, pero la más usada es 9600, pero depende de los relojes de los dispositivos para establecer la mejor frecuencia. Solamente debe ser la misma frecuencia en ambos dispositivos para evitar malas conexiones y lecturas con ruido. Una ventaja de este protocolo es su asincronía para comunicarse con el otro dispositivo (Fang & Chen, 2011).

#### *2.4.2.4 CAN*

CAN (Controller Area Network), es un protocolo de comunicación alámbrica que maneja multi-maestro para su comunicación. Puede soportar hasta 110 dispositivos en el mismo canal de comunicación, en este protocolo, la frecuencia cambia en la distancia que haya entre los dispositivos, por ejemplo, si un componente se encuentra cerca de otro, la frecuencia será alta. Para aplicaciones en tiempo real se utiliza comúnmente el protocolo CAN debido que no requiere nada especial para el medio de transmisión (Wang & Guo, 2009).

### **2.4.3 Capa de Aplicación**

Para sistemas de IoT se encuentran diferentes tipos de protocolos de mensajería, los cuales se hablará a continuación, con características básicas de cada protocolo además de algunas ventajas y desventajas de cada uno.

#### *2.4.3.1 HTTP*

HTTP (proviene de las siglas de inglés Hyper Text Transport Protocol), es predominantemente un protocolo de mensajería web, el cual fue publicado como estándar en 1997, desarrollado por Tim Berners. HTTP soporta solicitudes y respuestas de la arquitectura Web RESTful, también utiliza Identificadores universal de recursos (URI) en lugar de topics. El Servidor envía datos a través del URI y el cliente recibe los datos a través de una URI particular. En este protocolo el tamaño predeterminado para los encabezados y el mensaje ya que esto dependerá del servidor web o de la programación directamente. HTTP utiliza TCP como protocolo de transporte por default y TLS/SSL para seguridad. HTTP es un estándar de mensajería web que ofrece varias características, como, conexiones persistentes, canalización de solicitudes, y codificación de transferencia fragmentada (Naik, 2017).

#### *2.4.3.2 MQTT*

MQTT (Message Queuing Telemetry Transport) es uno de los protocolos de comunicación máquina a máquina (M2M) más antiguo, se introdujo en 1999, fue desarrollado por Andy Stanford-Clark y Arlen Nipper. Es un protocolo de mensajería de publicaciones y suscripciones diseñado para comunicaciones M2M de bajo consumo para redes restringidas. MQTT tiene clientes, corredores (brokers), temas (topics), los clientes publican mensajes hacia el corredor de MQTT, donde están los suscriptores por otros clientes o puede ser retenido para una futura suscripción, cada mensaje es publicado hacia un destino, en este caso como un tema. Los clientes pueden suscribirse a múltiples temas y recibir cada mensaje publicado en cada tema. MQTT requiere de un título fijo de 2 bytes con una carga útil de mensaje máximo de 256 MB, usa TCP como protocolo de transporte y TLS/SSL para seguridad, además de tener 3 niveles de Servicio

de Calidad (QoS), para entregas confiables para los mensajes (MQTT.org, 2020; Naik, 2017).

## **2.5 Servidores en la nube.**

En la actualidad los proveedores de servicios de la nube han creado su apartado de IoT con una gran diversidad de servicios que ofrecen de su parte. Grandes empresas como Google, Amazon, IBM, entre otras, han creado sus propios ecosistemas y con funciones únicas. En esta parte del capítulo se explicará las opciones que ofrece cada nube de Google, Amazon, IBM como nubes privadas y una de código abierto que es Node-RED.

La primera nube privada es la de Google, Google Cloud IoT, es un conjunto de herramientas para conectar, procesar, almacenar y analizar los datos tanto como en local como en la nube. Esta plataforma se compone de servicios en la nube escalables y totalmente gestionados, una serie de software integrada con funciones de aprendizaje automático para los recursos de computación locales para satisfacer las necesidades de Internet de las cosas. Los usos más comunes para esta plataforma son, Mantenimiento predictivo, seguimiento de recursos en tiempo real, logística y gestión de la cadena de suministro, y ciudades y edificios inteligentes. Para lograr esto hace uso de funciones de su plataforma de Cloud, por ejemplo, Cloud Dataflow, Cloud Pub/Sub, Cloud IoT Core (para conexión de dispositivos y su administración, BigQuery, Cloud ML Engine (Machine Learning), entre otros (Google, 2020).

La segunda plataforma es la de Amazon AWS (Amazon Web Services), esta plataforma tiene su segmento de IoT igual que la anterior, que ofrece servicios para la industria, consumidor y soluciones comerciales. Algunos casos de usos que ofrece son, en el área industrial, para la predicción de calidad y de mantenimiento de maquinaria, el monitoreo remoto de operaciones. En vivienda Inteligentes, o casas inteligentes, aplicaciones para automatizar la casa, seguridad y monitoreo de la casa. El área comercial es la creación de aplicaciones comerciales para el monitoreo de tráfico, seguridad pública y monitoreo de la salud. Servicios de AWS IoT que ofrece son sistemas operativos para microcontroladores, software para utilizar sus herramientas de manera

local, administración de dispositivos conectados a la nube para monitoreo, registros de seguridad y organización a gran escala de manera remoto, análisis de la información recabada de los dispositivos, entre otras más aplicaciones y funcionalidades que se pueden encontrar en esta plataforma (Amazon Web Services, 2020).

La nube de IBM tiene una gran ventaja la cual es Watson IoT. Watson IoT es una plataforma desarrollada por IBM, en la cual utiliza la inteligencia artificial en la nube, que ayuda a conectar de manera fácil los dispositivos, recopilar y procesar los datos de la empresa que ya tenga el servicio de IBM Cloud, además ofrece servicios para la visualización y análisis (IBM, 2020b). IBM Cloud también contiene su apartado de Internet de las cosas, el cual tiene APIs para conectar sus aplicaciones con los dispositivos o adaptadores, se pueden comunicar por MQTT y HTTP para enviar los datos de los dispositivos a la nube. Para la administración, análisis e interpretación de la información se pueden crear aplicaciones dentro de su nube con estos propósitos. Un gran punto para resaltar de esta plataforma es Watson, ya que es una inteligencia artificial líder en el mercado, con lo cual se puede aprovechar para que esta inteligencia artificial utilice los datos para generar un aprendizaje continuo, y con ello sugerir estrategias para un crecimiento, soluciones abiertas y flexibles, seguridad confiable, todo realizándose de manera remota y prácticamente en tiempo real (IBM, 2020a).

Por último, se encuentra Node-RED que está basado en Node.js, el cual es de código abierto, tiene compatibilidad ya sea en computadores o dispositivos comerciales (como Raspberry Pi, Android), y uso en nubes comerciales. La ventaja principal es su código abierto, con lo cual tiene un gran apoyo por parte de la comunidad para el aporte de funcionalidades y asistencia para aclarar dudas o fallas en los programas. Node-RED es una herramienta de programación para integrar dispositivos, APIs y servicios en línea de maneras nuevas, provee de un editor basado en navegación el cual facilita la manera de programar e interactuar con el software, ya que se refleja de manera de conexiones de flujo con los nodos de programación que tiene. Por otra parte, puede desplegar información en Internet y cómo almacenar datos, todo esto funcionando como una nube local el cual no genera ningún costo ya que es de código abierto (Open JS Foundation, 2020).



## Capítulo 3. Metodología.

En este capítulo se hablará sobre los equipos, métodos y pruebas utilizados para el desarrollo del dispositivo adaptador para IoT. Todas las pruebas se realizaron en las instalaciones de CETYS Universidad Campus Mexicali, con el apoyo y equipo de esta. Los principales equipos fueron los componentes internos del vehículo, el mismo vehículo, computadoras de escritorio, impresoras 3D, softwares open-source, como específicos para dispositivos que así lo requerían.

En el campus Mexicali, se tiene una celda de trabajo para el desarrollo del vehículo autónomo, en el cual se puede encontrar todo lo relacionado para realizar alguna implementación y/o modificación en piezas en chasis, así como de componentes del vehículo como de electrónica y sus respectivos programas dentro de las computadoras que se encuentran dentro de la celda de trabajo. En esta celda trabajan desde alumnos de profesional hasta doctores investigadores dentro del mismo proyecto.

Para el muestreo se utilizaron los siguientes componentes electrónicos:

- National Instruments roboRIO.
- ATmega 2560 microcontrolador.
- ATmega 328 microcontrolador.
- WeMo D1 mini microcontrolador.
- Raspberry Pi 3.
- Victor SPX (controladores de motores).
- Optical Encoders.
- Sensores Ultrasónicos.

En el área de software se utilizaron los siguientes:

- National Instruments LabVIEW 2019® (32 bits).
- Arduino IDE.
- Raspbian.
- Node-RED.

Metodología se dividió en 3 secciones, donde cada etapa aportará una o más características para Módulo Adaptador, además se obtendrán resultados que ayudarán a la siguiente etapa para la preparación de sus pruebas:

### 3.1 Primera Etapa.

Para la realización de las pruebas se desarrolló la siguiente metodología (Fig. 7),

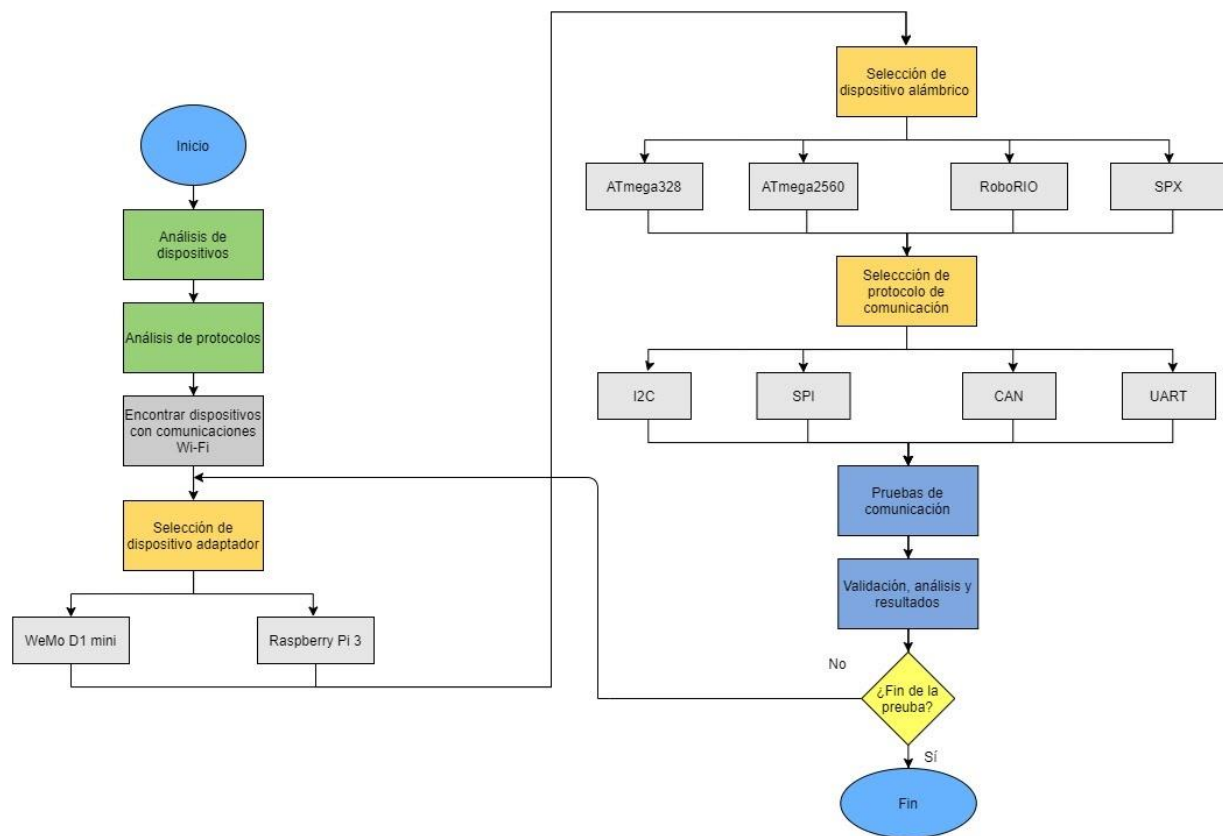


Fig. 7. Diagrama de flujo para las pruebas de los componentes internos del vehículo.  
Fuente: Elaboración propia.

El objetivo de la primera etapa es seleccionar el dispositivo adaptador y su protocolo de comunicación de manera interna del vehículo autónomo. Para ello, primero será analizar de manera individual a cada uno de los componentes electrónicos que se encuentran dentro del vehículo interno, para identificar protocolos de comunicación en que pueden trabajar, las especificaciones generales de los componentes, como, puertos de entradas y salidas, voltajes, alimentación, si contienen módulos de comunicación Wi-Fi.

Posteriormente de analizar cada componente, se realizará un segundo análisis para cada protocolo de comunicación que se hallaron de los dispositivos electrónicos. En este análisis se enfocará para la obtención de características específicas, como, velocidades de transferencias, tipo de datos que puede ser enviados, tipo de comunicación, número de dispositivos que pueden comunicarse a un solo maestro o canal.

Una vez terminados con los análisis a los componentes electrónicos y a los protocolos, se podrá avanzar a las pruebas de comunicación. Todos los componentes con módulos de comunicación Wi-Fi se considerarán como candidatos a dispositivos adaptadores, y los demás sin este módulo serán dispositivos para adaptarse por medio de algún protocolo de comunicación en común con el dispositivo adaptador. La prueba consiste en seleccionar un candidato a dispositivo adaptador, luego seleccionar un dispositivo por adaptar, y, por último, seleccionar un protocolo de comunicación en común entre ambos dispositivos, para enviar los datos de prueba de manera bidireccional.

Las pruebas deberán realizarse para cada candidato a dispositivo adaptador con todos los dispositivos para adaptar, por todos los protocolos de comunicación en común. Los resultados que se buscan son: los tiempos de transferencias, conexiones exitosas, porcentaje de precisión de los datos de prueba, y los tipos de datos que pueden ser enviados.

### 3.2 Segunda Etapa.

El objetivo de la segunda etapa es realizar pruebas de conexión del dispositivo adaptador seleccionado de la primera etapa con los servidores en Internet. Para esta etapa, ya se deberá tener un dispositivo adaptador con la habilidad de conectarse a Internet.

La siguiente figura muestra la segunda etapa de la metodología:

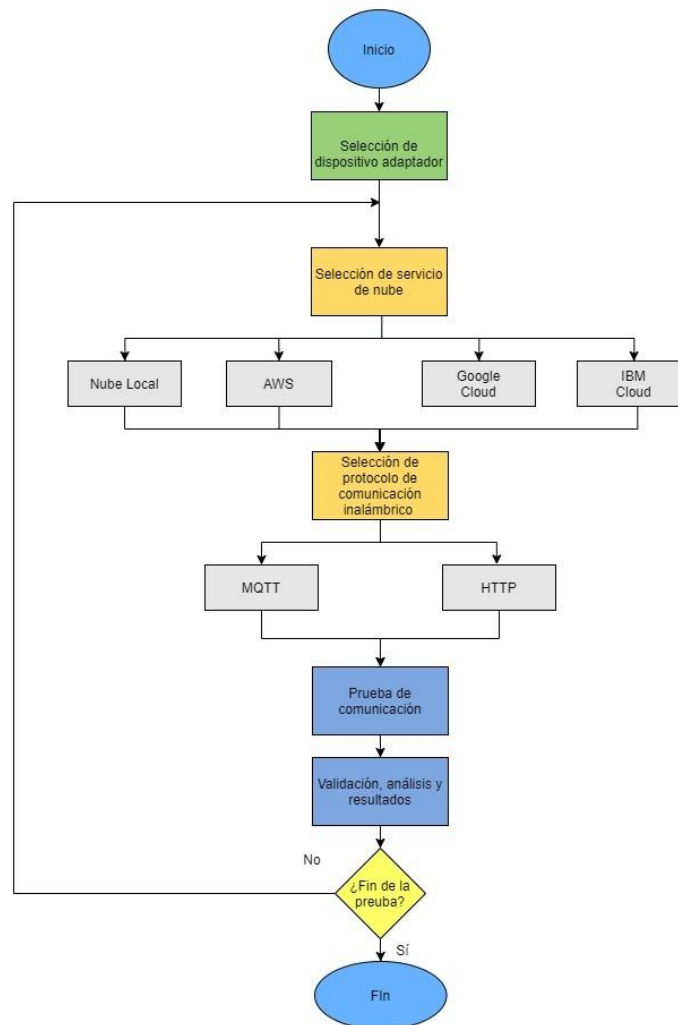


Fig. 8. Diagrama de flujo para las pruebas de las nubes y protocolos de la capa de aplicación.  
Fuente: Elaboración propia.

En esta figura explica la comunicación del dispositivo adaptador con la nube local que se tiene dentro del campus para la experimentación del proyecto del vehículo autónomo, además de realizar pruebas con nubes privadas como Amazon Web Services (AWS), Google Cloud y IBM Cloud.

Para la nube local se utilizará una Raspberry Pi 3 con el sistema operativo Raspbian, con el software Node-RED, este último, servirá como servidor en la nube, además de administrar la base de datos y la administración del tablero interactivo en la nube. El dispositivo adaptador se comunicará con la nube local por medio de los protocolos de comunicación Message Queuing Telemetry Transport (MQTT) y Hypertext Transfer Protocol (HTTP), para una comunicación bidireccional. Para las nubes privadas se seleccionarán ambos de protocolos de comunicación previamente mencionados para la realización de las pruebas.

Para empezar esta segunda parte de la metodología es la selección del dispositivo adaptador con conexión a Wi-Fi. Una vez seleccionado, el siguiente paso es seleccionar una nube a la cual se desea conectar, para esto se debe realizar una configuración inicial en el cual se puedan otorgar los permisos necesarios para la conexión éxito con la nube, debido que algunas requieren llaves de acceso o usuarios y contraseñas.

Posteriormente, será la selección del protocolo de comunicación inalámbrica, entre MQTT y HTTP. Una vez realizada la conexión con la nube y el protocolo de comunicación inalámbrica configurada, se realizará las pruebas de comunicación., de igual a la primera etapa de la metodología, se enviará y recibirá información para obtener los resultados, que se categorizaran de la siguiente manera, mensaje enviado por parte del emisor, mensaje recibido por parte del receptor, regresar la información al emisor y comprobación, la información enviada será una cada de String. Por último, se calcula el porcentaje de precisión de los mensajes.

Todas las pruebas deberán realizarse para cada nube indistintamente sea privada o local, además de realizar las pruebas por ambos protocolos de comunicación inalámbricas MQTT y HTTP con el dispositivo adaptador.

### **3.3 Tercera Etapa.**

En esta etapa se realizarán las pruebas en un entorno real, donde el dispositivo adaptador se encontrará dentro del vehículo, además en la nube local se tendrá el tablero interactivo con información en tiempo real sobre el vehículo. El objetivo de esta última etapa es corroborar la confiabilidad de mostrar, y almacenar los datos recabados por parte del vehículo; probar y ejecutar los comandos desde el tablero interactivo. Además de comprobar si la red inalámbrica del campus posee la capacidad para la implementación del vehículo dentro del campus. Con ello se podrá identificar qué zonas de las instalaciones carecen de cobertura y segmentar el ancho de banda específicamente para el proyecto del vehículo autónomo.

Para la realización de esta prueba se usará la la nube que tenga mejores ventajas y se acomode mejor a las necesidades que ocupa el proyecto del Vehículo Autónomo de CETYS Universidad, el cual se comunicará por el protocolo más conveniente para esta tarea. En esta etapa como resultados se deberá comunicar todos los datos que se tengan del vehículo hasta el momento y mandarlos a la nube, para posteriormente desplegarlo en un tablero interactivo con botones interactivos que puedan mandar comando al vehículo, como un botón de paro de emergencia, o de reanudar actividades.

La información necesaria para esta etapa que deberá desplegar son las velocidades de los motores (valores absolutos), desplazamientos por cada llanta, valores de presencia con los sensores ultrasónicos para 4 áreas principales del vehículo, por último, valores de consumo de energías (como voltajes, amperes y watts). En la siguiente figura se explicará el flujo de las interacciones y pasos que deberá tener esta tercera etapa:

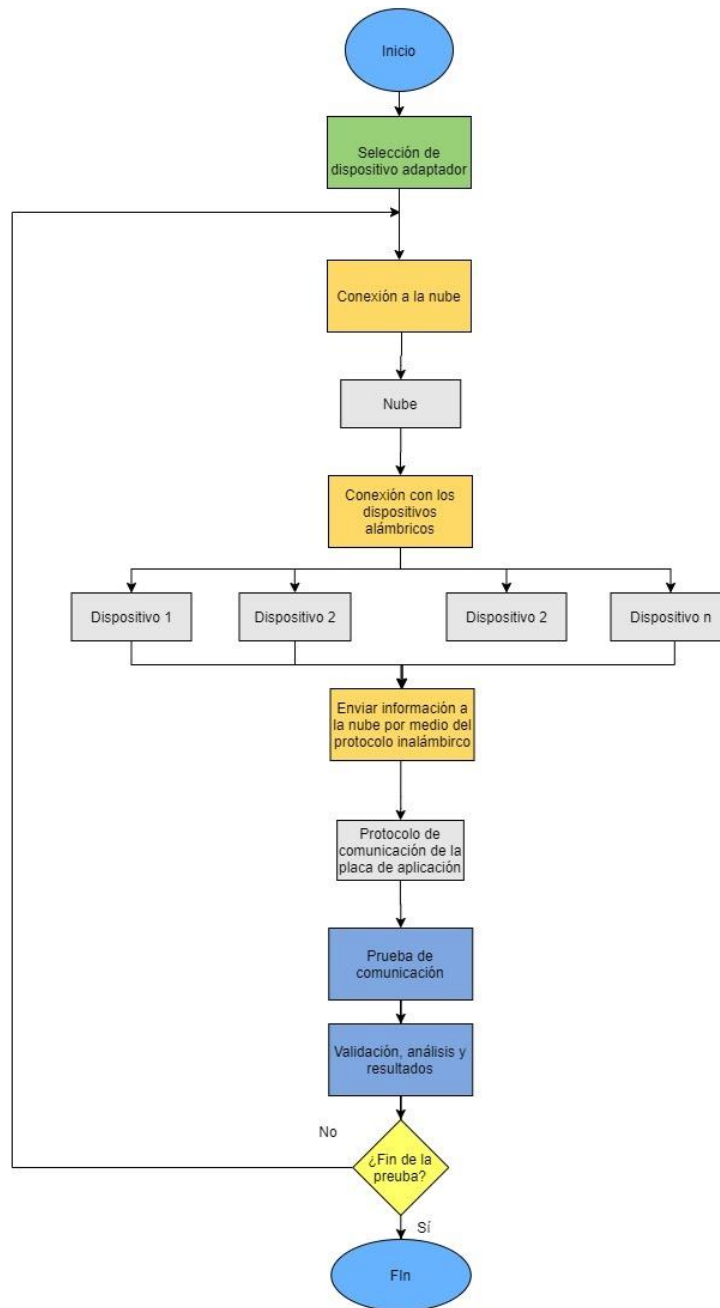


Fig. 9. Diagrama de flujo para el módulo adaptador dentro del vehículo.  
Fuente: Elaboración propia.

## Capítulo 4. Resultados.

En este capítulo se habla de los resultados obtenidos en cada etapa previamente explicada en el capítulo 3 de la metodología. Los resultados de cada etapa se encuentran separados describiendo el procedimiento de cada prueba.

### 4.1 Primera Etapa

En esta primera etapa se analizaron los dispositivos para obtener sus características sobre sus protocolos de comunicación tanto inalámbricos y alámbricos que pueden utilizar. Con estas características se perfilaron como prospectos a dispositivos adaptadores al Raspberry Pi y al Wemo D1. A parte de esas características se relacionaron los protocolos de comunicación alámbricos para poder observar sus compatibilidades de comunicación. En la siguiente tabla 1 se puede observar que protocolos puede utilizar cada dispositivo y las compatibilidades que tiene cada protocolo.

	Ni roboRIO	Atmega 328		Raspberry pi 3	WeMo D1 mini	Victor SPX
I2C	Yes	Yes	Yes	Yes	Yes	No
SPI	Yes	Yes	Yes	Yes	Yes	No
UART	Yes	Yes	Yes	Yes	Yes	No
CAN	Yes	No	No	No	No	Yes

Tabla 1. Resultados de los protocolos alámbricos por cada dispositivo interno del vehículo.

Fuente: Elaboración propia.

Con estos resultados se continuó con las pruebas de los protocolos de comunicación alámbricas que cuentan los dos dispositivos con módulos de comunicación inalámbricos. Las pruebas que se realizaron se utilizaron todos los dispositivos que se encontraban dentro del vehículo; los dispositivos contaban con los mismos protocolos a excepción de los Victor SPX (que se retiraron de las pruebas ya que no se podían comunicar directamente con los dos dispositivos prospectos). Las pruebas se realizaron utilizando un protocolo en común, donde se utilizaba un lenguaje de programación que se adecuara mejor a cada dispositivo, para los casos de los ATMegas se utilizó el IDE



de Arduino, para el Wemo D1 igualmente se usó el IDE de Arduino, en el roboRIO se utilizó el programa proveniente por parte de National Instruments (LabVIEW), en el caso de Raspberry Pi se usó los lenguajes de Python.

Una vez establecida la conexión se realizó la prueba, en la cual se envió el tipo de dato que soportará el protocolo universalmente que en este caso eran byte (int) o String. El dispositivo 1 enviaba un dato al dispositivo 2 por el protocolo establecido para la prueba, el dispositivo 2 el cual recibía el dato, lo regresaba por el mismo protocolo, en esta prueba se imprimía el dato enviado y recibido por el dispositivo 1, y el tiempo de respuesta en s en cada iteración, se realizaron 1000 para cada protocolo y se obtuvieron los resultados de la tabla 2, donde se puede apreciar que el protocolo SPI es el más rápido pero con mayor errores en la transferencias de datos, el protocolo I2C fue el segundo más rápido y con 100% de confiabilidad que todos los datos llegaron correctamente. Por último, el protocolo fue UART, el cual fue el único que soporto enviar cadenas largas de caracteres sin importar el lenguaje o dispositivo, presentó un mayor retardo en el envío de información a comparación de los demás protocolos, pero se debe tener en cuenta que este protocolo envió cadenas de caracteres y no simplemente un byte, y su única falla era siempre la primera interacción después de eso no volvía a fallar.

Protocolos	Promedio de tiempo de respuesta (s)	Pruebas realizadas	Promedio de datos correctos (%)	Conexiones correctas (%)	Tipo de dato
SPI	0.000054701699	1000	96.30%	100.00%	Byte(int)
I2C	0.000517495327	1000	100.00%	100.00%	Byte(int)
UART	0.007411107	1000	99.90%	100.00%	String

Tabla 2. Resultados de las pruebas por cada protocolo de comunicación alámbrica.

Fuente: Elaboración propia.

## 4.2 Segunda Etapa

En la segunda etapa se realizaron pruebas con los protocolos de comunicación más usados en Internet of Things, los cuales son MQTT y HTTP. Para ello se investigó si en todas las nubes que se querían realizar las pruebas tanto comerciales (AWS, Google Cloud, IBM Cloud) como locales (Node-RED) soportan estos dos protocolos de

comunicación, en la siguiente tabla se puede observar que protocolos son soportados por cada nube. (tabla 3).

Protocolos	AWS	Google Cloud	IBM Cloud	Node-RED
HTTP	Si	Si	Si	Si
MQTT	Si	Si	Si	Si

Tabla 3. Resultados de los protocolos de la capa de aplicación por cada nube.  
Fuente: Elaboración propia.

Una vez analizado que en todas las nubes son compatibles ambos protocolos se realizaron, pruebas para cada uno de los protocolos. Las pruebas que se realizaron fueron distintas debido a que no son distintos los comandos y arquitecturas de ambos protocolos. En la fig. 10 se muestra la manera en que se realizó la prueba para el protocolo de HTTP. En esta prueba se utilizaron los dispositivos Wemo D1 y el Raspberry Pi, los servidores fueron montados en el Raspberry Pi y los Wemos como los dispositivos que se comunicaban a través de los protocolos. El primer Wemo D1 utilizaba el comando POST para cierta dirección, y solicitaba la información con el comando GET a otra dirección diferente, mientras el segundo Wemo D1 usó las direcciones al revés. Entonces se pudo obtener los siguientes resultados que se muestran en la tabla 4 (HTTP).

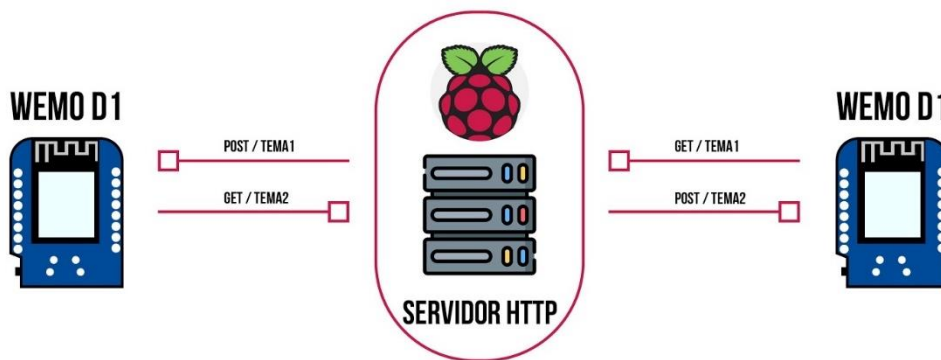


Fig. 10. Diagrama de conexión para la prueba del protocolo HTTP.  
Fuente: Elaboración propia.

En el caso del protocolo MQTT se tuvo que hacer otro método de prueba debido que este protocolo soporta devoluciones de llamadas. Lo cual permitió, que se realizará la prueba similar a los protocolos alámbricos que también soportan este tipo de comandos. Por lo tanto, el Fig. 11 se puede observar el flujo de la prueba para este

protocolo, el cual igualmente se utilizó el Raspberry Pi como un MQTT bróker, y dos Wemo D1 para comunicarse entre sí, y esperando que el primer Wemo D1 publicará en su tema para que el segundo dispositivo pudiera recibirlo y publicar la información al segundo tema, donde el primer Wemo D1 está suscripto. En la siguiente tabla (tabla 4) MQTT se muestran los resultados de esta prueba.

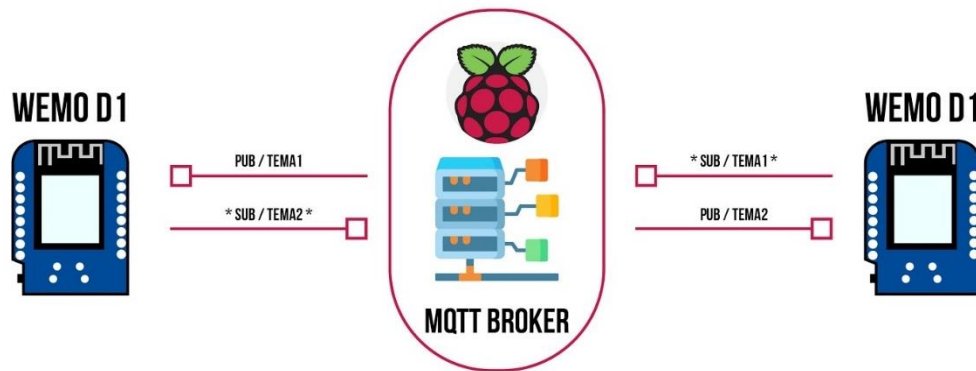


Fig. 11. Diagrama de conexión para la prueba del protocolo MQTT.  
Fuente: Elaboración propia.

Protocolos	Promedio de tiempo de respuesta (ms)	Pruebas realizadas	Promedio de datos correctos (%)	Conexiones correctas (%)	Tipo de dato
HTTP	111.0119389	2094	84.53%	100.00%	Byte(int)
MQTT	33.8862916	2524	100.00%	100.00%	Byte(int)

Tabla 4. Resultados de los protocolos de comunicación de la capa de aplicación.  
Fuente: Elaboración propia.

En la tabla anterior se puede observar que el protocolo MQTT fue el más acertado y rápido entre los dos protocolos, además de tener la devolución de llamadas (callbacks) para los dispositivos que están registrados en el tema.

Para las pruebas con las nubes se utilizó el protocolo MQTT, debido al método de devolución de llamada, además se utilizaron las herramientas, funciones, y aplicaciones que cada nube ofrecía para poder utilizar mejor sus recursos, con sus apartados de IoT. Para esta prueba se tuvo que utilizar dos Wemo D1 para cada prueba, además en las nubes comerciales se registraron los dispositivos, a parte se flashearón para que pudieran tener las llaves y certificados de cada uno y para cada nube. Además, para la nube local (Node-RED) se utilizó el bróker anteriormente ya que corre ambos

servicios el Raspberry Pi. En las siguientes figuras (12-14) se describe el flujo en que se realizó para cada nube de esta prueba.

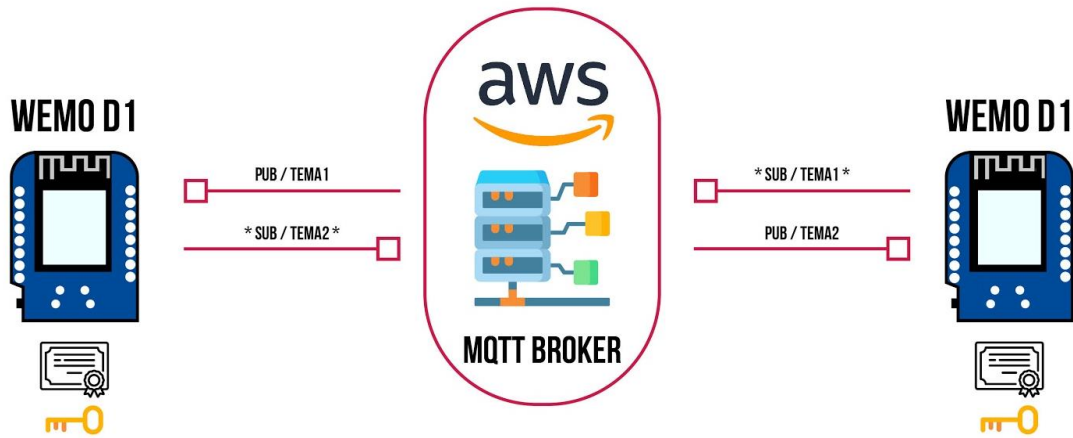


Fig. 12. Diagrama de conexión para la prueba para nube de AWS con el protocolo de comunicación MQTT.  
Fuente: Elaboración propia.

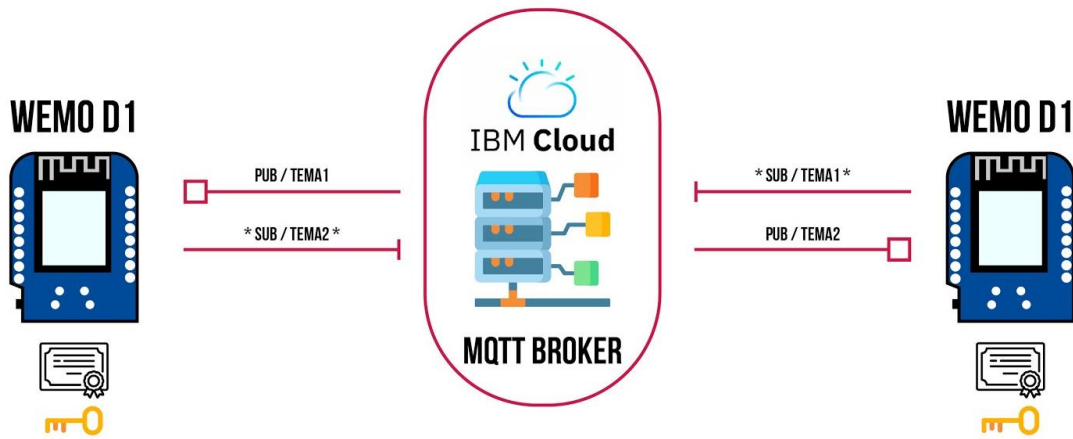


Fig. 13. Diagrama de conexión para la prueba para nube de IBM Cloud con el protocolo de comunicación MQTT.  
Fuente: Elaboración propia.

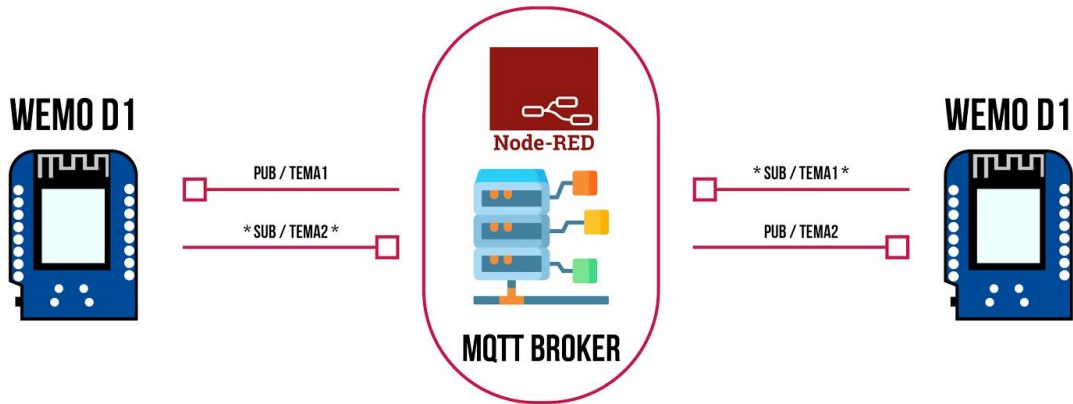


Fig. 14. Diagrama de conexión para la prueba para nube de Node-RED con el protocolo de comunicación MQTT.  
Fuente: Elaboración propia.

Como resultados se obtiene los siguientes datos que se muestran en la tabla 5 nubes

Nubes	Promedio de tiempo de respuesta (ms)	Pruebas realizadas	Promedio de datos correctos (%)	Conexiones correctas (%)	Tipo de dato
AWS	99.1866931	1007	100.00%	100.00%	Json
Node-RED	30.3266842	5715	100.00%	100.00%	Json
Google Cloud	NA	NA	NA	NA	NA
IBM Cloud	0.55201*	600	100.00%	100.00%	Json

Tabla 5. Resultados de las pruebas nubes comerciales y locales.  
Fuente: Elaboración propia.

En la tabla 5 son los resultados de las pruebas con las nubes, no en todas las nubes se pudieron realizar las pruebas y en otras hubo limitaciones por parte de la nube, en todas las pruebas se utilizó el mismo formato Json con datos variables en cada iteración. A grandes rasgos se puede rescatar que Node RED fue la mejor nube entre todas por los tiempos de respuestas, algo también para rescatar son las conexiones y los datos correctos que no hubo ningún error en las nubes que se pudieron realizar las pruebas.

### 4.3 Tercera Etapa

Como parte de la tercera etapa se replicó la arquitectura real del sistema electrónico del vehículo autónomo, con el objetivo de realizar las pruebas de esta etapa en el sistema réplica en vez del vehículo autónomo. Esto debido a que por la situación de la contingencia global por la pandemia del virus COVID-19, por disposición oficial, el acceso a las instalaciones de CETYS Universidad campus Mexicali fue restringido, dificultando los procesos de experimentación en las instalaciones.

De esta manera las pruebas fueron realizadas sobre la réplica del sistema base del vehículo autónomo. El objetivo de esta última etapa es analizar la confiabilidad de almacenar y mostrar los datos recabados por parte del vehículo, y además probar los comandos de control desde el tablero interactivo.

Para esta etapa se utilizó el protocolo MQTT para comunicarse con la nube, la cual fue Node-RED, en el cual también está localizado el tablero interactivo. Para la comunicación entre los dispersivos internos del vehículo con el módulo adaptador se utilizó el protocolo UART para poder enviar datos tipo String. El dispositivo adaptador recolectaba la información de los dispositivos, haciendo primeramente un requerimiento de información para cada dispositivo, una vez recolectada la información de todos los dispositivos, el mismo dispositivo adaptador da el formato Json a la información para enviarlos a los temas MQTT correspondientes (Desplazamiento, Velocidades, Sensores, Consumo energético).

Como resultados de las pruebas se obtuvo lo siguiente:

Nube	Promedio de tiempo de respuesta (ms)	Pruebas realizadas	Promedio de datos correctos (%)	Conexiones correctas (%)	Tipo de dato
Node-RED	38.0000000	10000	100.00%	100.00%	Json

Tabla 6. Resultados de la prueba del módulo adaptador con la nube Node-RED.

Fuente: Elaboración propia.

El tiempo se consideró desde que el dispositivo adaptador solicita al primer dispositivo la información y se termina hasta que recibe la notificación del método de la devolución de llamada del último tema que publicó el mismo dispositivo adaptador, no

hubo errores de comunicaciones ni datos erróneos, en la siguiente figura se verá el tablero interactivo que se utilizó para esta prueba.



Fig. 15. Captura de pantalla del tablero interactivo con datos del vehículo autónomo.

Fuente: Elaboración propia.

En la figura 16 se muestran las conexiones y comunicación entre dispositivos y nube para llevar a cabo esta prueba.

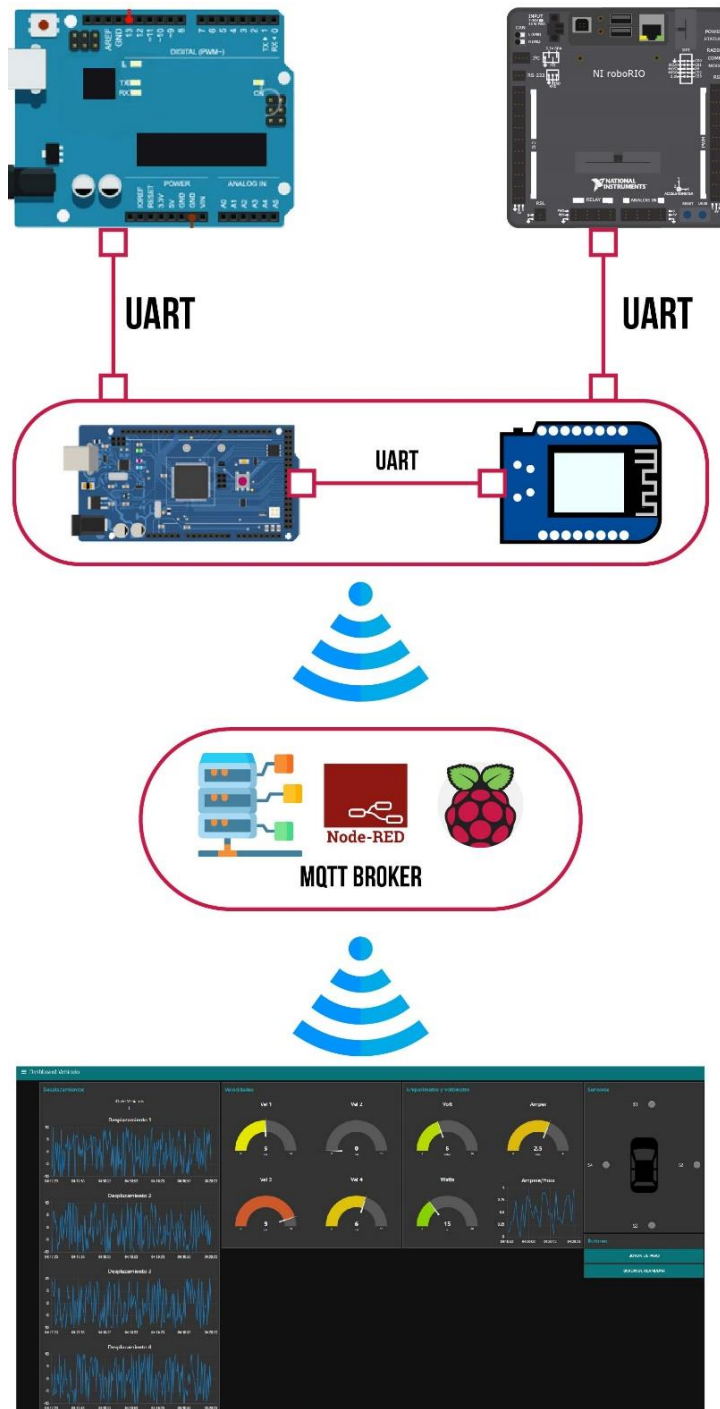


Fig. 16. Diagrama de conexión para la prueba del módulo adaptador para la comunicación de los componentes internos del vehículo autónomo a la nube de Node-RED.  
Fuente: Elaboración propia.



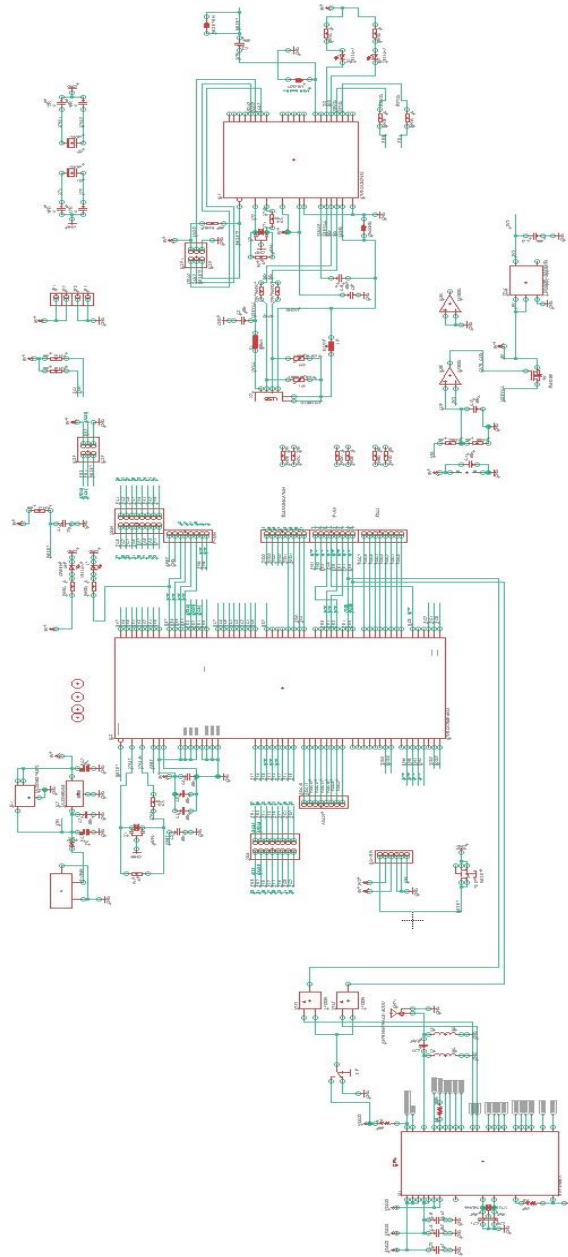


Fig. 17. Diagrama Esquemático del Módulo Adaptador.  
Fuente: Elaboración propia.

Adicionalmente se diseñó un circuito prototipo integrando todos los componentes electrónicos de ambos dispositivos que fueron seleccionados para trabajar como Módulo Adaptador. En la siguiente imagen se muestra un diagrama esquemático con las conexiones necesarias para su funcionamiento, se diseñó para programar los dispositivos de manera separada y que se comuniquen posteriormente como se mostró en este trabajo.

## Capítulo 5. Conclusiones.

Se hablará primeramente de las conclusiones, observación y discusiones por cada etapa separada, y al final habrá una sección general como oportunidades de mejoras, recomendaciones y líneas de investigación futuras.

En la primera etapa se decidió que UART sería el protocolo de comunicación con interfaz alámbrico, debido que este permite el intercambio de información tipo String desde cualquier dispositivo que se probó en la experimentación. Además, que su único error fue su primera iteración, posteriormente de esa iteración todos los datos se enviaban y recibían correctamente. El protocolo CAN se retiró de las pruebas debido que solamente se encontraba en los dispositivos roboRIO y Victor, los cuales se comunicaron por medio de señales PWM. Los dispositivos Raspberry Pi y Wemo D1 se consideraron para dispositivos adaptadores debido a que cuentan con módulos de comunicación inalámbricas (Wi-Fi).

Con la información obtenida de esas pruebas se optó como dispositivo adaptador al Wemo D1 más la adición de un ATmega 2560, ya que tiene cuatro canales UART, además de tener entradas y salidas para comunicarse directamente con los actuadores o sensores que pueda tener el sistema en este caso el vehículo autónomo de CETYS Universidad.

Para la segunda etapa se seleccionó el protocolo de comunicación de la capa de aplicación, el cual fue MQTT por su método de devolución de llamada, el cual envía el mensaje a los dispositivos suscritos al temas que se acaba de hacer la publicación, lo cual evita realizar el método GET del protocolo HTTP, el cual se tuvo que realizar la prueba evitando comparar directamente dentro del dispositivo si la información era correcta, ya que esto se revisó posteriormente de la prueba para calcular los tiempo para que el segundo dispositivo viera reflejado el cambio de estado. A parte que MQTT tuvo el 100% de datos correctos y ser 3 veces más rápido que su rival.

En la parte de las nubes se tuvo dificultades debido a que se utilizaron las herramientas que estas mismas proveen para IoT, las cuales soportan los protocolos

MQTT y HTTP para mensajería, pero como se mostró en la prueba pasada todo se basó en el MQTT por sus beneficios. Las dificultades surgieron en las nubes de Google y de IBM, en la nube Google no soporta nativamente dispositivos como Wemo, sino como Raspberry Pi, no se descartó se intentó con librerías de personas de la comunidad de Github, pero sin éxito debido que las librerías eran para específicos dispositivos, por lo tanto, no se pudo realizar las pruebas dentro de los Wemos. Por otro lado, la nube de IBM puede generar certificados y llaves para dispositivos y APIs, el único inconveniente con los dispositivos dentro de esta nube no tiene la accesibilidad de suscribirse a temas, pero se pueden mandar ciertos comandos específicos que cuenta la nube, por lo tanto, se pierde la ventaja de tener el método de devolución de llamada.

Con respecto a los resultados de la tabla 5, el promedio de tiempos de respuesta de en el caso de IBM el tiempo es la diferencia que hay de un mensaje enviado a otro por el mismo dispositivo el cual sólo confirma que fue enviado, pero no el tiempo que se vio reflejado en el servidor. En el caso de la nube de Amazon (AWS), este debía de tener un retraso de 2 segundos entre mensaje y mensaje para evitar errores de comunicaciones los cuales no se consideraron, solo se tomó el tiempo desde que se envió el dato y se recibe el mensaje del tema suscrito. Se seleccionó Node-RED ya que fue el más rápido, y no era necesario tener retrasos para enviar los mensajes, además que es la opción más barata porque se puede montar desde el Raspberry Pi que se puede aprovechar los recursos que serán removidos debido al dispositivo adaptador.

En la tercera etapa se puso en prueba el módulo adaptador integrado por el ATMega 2560 y el Wemo D1, el cual se comunicó con 3 dispositivos a la vez por medio del protocolo UART, y a la nube por medio de MQTT. Las pruebas se realizaron en un entorno donde la señal de Internet no tenía ninguna interferencia, algo no se puede asegurar en el campus sin pruebas en ese entorno real para ver las deficiencias que pueda tener el sistema. Para el tablero interactivo contó con todos los parámetros que el vehículo tiene hasta el momento incluidos, además de agregar botones de comandos, donde el principal era el botón de emergencia. A fin de cuentas, se pudo obtener un dispositivo adaptador que fuera capaz de comunicarse con los dispositivos internos del

vehículo y a la nube tanto comercial como locales obteniendo resultados aceptables basándose en las velocidades de respuestas y la confiabilidad de las pruebas.

Se implementó satisfactoriamente el sistema embebido como dispositivo adaptador para IoT, el cual redujo el número de dispositivos en la arquitectura del vehículo autónomo y optimizando los recursos de los dispositivos internos, con lo que se redujo el consumo de energía.

En todo caso, como recomendaciones, se propone realizar pruebas del dispositivo dentro del campus para evaluar resultados y comportamientos que puedan ayudar a la cobertura de la señal, si tiene una respuesta más extensa o si mantendrá sus tiempos actuales.

Se desarrolló un dispositivo adaptador capaz de comunicarse con todos los dispositivos internos del vehículo autónomo de CETYS Universidad, y a la nube para guardar los datos y generar un tablero interactivo con comandos y el despliegue de información en tiempo real.

El módulo adaptador desarrollado es de bajo costo, utiliza dispositivos de código abierto, por lo tanto, posee una gran comunidad de soporte.

Finalmente, como futuras líneas de investigación, se buscará la robustez del diseño para su implementación en un entorno industrial y su comparación con dispositivos comerciales.

## Referencias.

- Adriano, J. D., Rosario, E. C. do, & Rodrigues, J. J. P. C. (2018). Wireless Sensor Networks in Industry 4.0: WirelessHART and ISA100.11a. *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, 924–929. <https://doi.org/10.1109/INDUSCON.2018.8627177>
- Alkar, A. Z., & Buhur, U. (2005). An internet based wireless home automation system for multifunctional devices. *IEEE Transactions on Consumer Electronics*, 51(4), 1169–1174. <https://doi.org/10.1109/TCE.2005.1561840>
- Amazon Web Services. (2020). *AWS IoT - Amazon Web Services*. Amazon Web Services, Inc. <https://aws.amazon.com/iot/>
- Apple Inc. (2020). *Watch*. Apple (México). <https://www.apple.com/mx/watch/>
- Basaca-Preciado, L. C., Orozco-Garcia, N. A., Terrazas-Gaynor, J. M., Moreno-Partida, A. S., Rosete-Beas, O. A., Rizzo-Aguirre, J., Martinez-Grijalva, L. F., & Ponce-Camacho, M. A. (2018). Intelligent Transportation Scheme for Autonomous Vehicle in Smart Campus. *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, 3193–3199. <https://doi.org/10.1109/IECON.2018.8592824>
- Buntz, B. (2017, septiembre 20). *The Top 20 Industrial IoT Applications*. IoT World Today. <https://www.iotworldtoday.com/2017/09/20/top-20-industrial-iot-applications/>
- Chen, B., Wan, J., Shu, L., Li, P., Mukherjee, M., & Yin, B. (2018). Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access*, 6, 6505–6519. <https://doi.org/10.1109/ACCESS.2017.2783682>

- Desai, P., Sheth, A., & Anantharam, P. (2015). Semantic Gateway as a Service Architecture for IoT Interoperability. *2015 IEEE International Conference on Mobile Services*, 313–319. <https://doi.org/10.1109/MobServ.2015.51>
- East Coast Polytechnic Institute. (2020). *A Brief History of Cloud Computing*. <https://www.ecpi.edu/blog/a-brief-history-of-cloud-computing>
- EUROTECH. (2020). *Everyware IoT - Edge to Cloud IoT architecture*. EUROTECH. <https://www.eurotech.com/en/products/iot#multi-service-iot-edge-gateways>
- Fang, Y., & Chen, X. (2011). Design and Simulation of UART Serial Communication Module Based on VHDL. *2011 3rd International Workshop on Intelligent Systems and Applications*, 1–4. <https://doi.org/10.1109/ISA.2011.5873448>
- FayerWayer. (2019). *El origen de: El Cómputo en la Nube*. FayerWayer. <https://www.fayerwayer.com/2012/01/el-origen-de-el-computo-en-la-nube/>
- Ferrigno, L., Paciello, V., & Pietrosanto, A. (2011). Experimental Characterization of Synchronization Protocols for Instrument Wireless Interface. *IEEE Transactions on Instrumentation and Measurement*, 60(3), 1037–1046. <https://doi.org/10.1109/TIM.2010.2060224>
- Fitbit. (2020). *Sitio oficial de Fitbit para smartwatches, pulseras de actividad, monitores deportivos y mucho más*. <https://www.fitbit.com/es/home>
- Glock, T., Pazmino Betancourt, V., Kern, M., Liu, B., Reib, T., Sax, E., & Becker, J. (2019). Service-Based Industry 4.0 Middleware for Partly Automated Collaborative Work of Cranes. *2019 8th International Conference on Industrial Technology and Management (ICITM)*, 229–235. <https://doi.org/10.1109/ICITM.2019.8710661>

- Goddard, W. (2019, junio 17). History of IoT: What It Is, How It Works, Where It's Come From, and Where It's Going. *ITChronicles*. <https://itchronicles.com/iot/history-of-iot-what-it-is-how-it-works-where-its-come-from-and-where-its-going/>
- Google. (2020). *Google Cloud IoT: Servicios de Internet de las cosas totalmente gestionados*. Google Cloud. <https://cloud.google.com/solutions/iot?hl=es>
- Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645–1660. <https://doi.org/10.1016/j.future.2013.01.010>
- Harwood, T. (2019, noviembre 1). *IoT Gateways | 2019 Comparison Guide to Hardware and Software Solutions*. Postscapes. <https://www.postscapes.com/iot-gateways>
- Hegde, Z. (2017, octubre 9). Eurotech expands its IoT gateway portfolio by introducing the new ReliaGATE 10-12 with integrated LTE connectivity. *IoT Now - How to Run an IoT Enabled Business*. <https://www.iot-now.com/2017/10/09/68662-eurotech-expands-iot-gateway-portfolio-introducing-new-reliagate-10-12-integrated-lte-connectivity/>
- Hu, L., Eichinger, J., Dillinger, M., Botsov, M., & Gozalvez, D. (2016). Unified Device-to-Device Communications for Low-Latency and High Reliable Vehicle-to-X Services. *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 1–7. <https://doi.org/10.1109/VTCSpring.2016.7504518>
- IBM. (2020a, mayo 13). *Conozca más acerca de la Internet de las cosas (IoT)*. <https://www.ibm.com/mx-es/internet-of-things>
- IBM. (2020b, agosto 10). *Watson IoT Platform—IBM Watson IoT*. <https://www.ibm.com/mx-es/business-operations/iot-platform>

- i-SCOOP. (2020). *IoT gateways and Industrial IoT gateways – usage and evolutions*. I-SCOOP. <https://www.i-scoop.eu/internet-of-things-guide/iot-gateways/>
- Knight, D. (2015, octubre 22). *Welcome to the commercial Internet of Things*. Network World. <https://www.networkworld.com/article/2991498/welcome-to-the-commercial-internet-of-things.html>
- Kruger, C. P., Abu-Mahfouz, A. M., & Hancke, G. P. (2015). Rapid prototyping of a wireless sensor network gateway for the internet of things using off-the-shelf components. *2015 IEEE International Conference on Industrial Technology (ICIT)*, 1926–1931. <https://doi.org/10.1109/ICIT.2015.7125378>
- Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business Horizons*, *58*(4), 431–440. <https://doi.org/10.1016/j.bushor.2015.03.008>
- Leens, F. (2009). An introduction to I<sup>2</sup>C and SPI protocols. *IEEE Instrumentation & Measurement Magazine*, *12*(1), 8–13. <https://doi.org/10.1109/MIM.2009.4762946>
- LEI Technology. (2020). *Commercial IoT Gateway*. Lanner. <https://www.lanner-america.com/products/iot-gateways/commercial-iot-gateway/>
- Matthews, K. (2019, mayo 30). *5 Benefits of Commercial IoT and Automation*. IoT for all. <https://www.ietfforall.com/5-benefits-commercial-iot-automation/>
- MaxBotix Inc. (2017). MB1240 XL-MaxSonar-EZ4. *MaxBotix Inc.* [https://www.maxbotix.com/Ultrasonic\\_Sensors/MB1240.htm](https://www.maxbotix.com/Ultrasonic_Sensors/MB1240.htm)
- Microchip Technology Inc. (2020a). *ATmega328—8-bit AVR Microcontrollers*. <https://www.microchip.com/wwwproducts/en/ATmega328>



- Microchip Technology Inc. (2020b). *ATmega2560—8-bit AVR Microcontrollers*.  
<https://www.microchip.com/wwwproducts/en/ATmega2560>
- MQTT.org. (2020). *MQTT - The Standard for IoT Messaging*. <https://mqtt.org/>
- Naik, N. (2017). Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. *2017 IEEE International Systems Engineering Symposium (ISSE)*, 1–7. <https://doi.org/10.1109/SysEng.2017.8088251>
- National Instruments. (2015). *NI roboRIO Specifications—National Instruments*. 10.
- Nest. (2020). *Create a Connected Home*. Nest. <https://www.nest.com/>
- Nordrum, A. (2016, agosto 18). *Popular Internet of Things Forecast of 50 Billion Devices by 2020 Is Outdated*. <https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated>
- NXP Semiconductors. (2020). *LS1021A-IoT Gateway Reference Design | NXP*.  
<https://www.nxp.com/design/designs/ls1021a-iot-gateway-reference-design:LS1021A-IoT>
- Open JS Foundation. (2020, enero 1). *Node-RED*. <https://nodered.org/>
- Qingping Chi, Hairong Yan, Chuan Zhang, Zhibo Pang, & Li Da Xu. (2014). A Reconfigurable Smart Sensor Interface for Industrial WSN in IoT Environment. *IEEE Transactions on Industrial Informatics*, 10(2), 1417–1425.  
<https://doi.org/10.1109/TII.2014.2306798>
- RASPBERRY PI FOUNDATION. (2015). *DATASHEET Raspberry Pi Compute Module 3+*.

[https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi\\_DATA\\_CM3plus\\_1p0.pdf](https://www.raspberrypi.org/documentation/hardware/computemodule/datasheets/rpi_DATA_CM3plus_1p0.pdf)

RASPBERRY PI FOUNDATION. (2020). *Buy a Raspberry Pi 3 Model B – Raspberry Pi*.

<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Savvides, A., & Srivastava, M. B. (2002). A distributed computation platform for wireless embedded sensing. *Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors*, 220–225.

<https://doi.org/10.1109/ICCD.2002.1106774>

Seeed Technology Co. (2020). *Grove IoT Commercial Gateway Kit*.

<https://www.seeedstudio.com/Grove-IoT-Commercial-Gateway-Kit-p-2724.html>

Tate, K. (2018, agosto 13). Commercial IoT is Different. *Rigado*.

<https://www.rigado.com/commercial-iot-is-different/>

US Digital. (2020). *E4T Miniature Optical Kit Encoder*. US Digital®.

<https://www.usdigital.com/products/encoders/incremental/kit/E4T>

VEX Robotics. (2020). *Victor SPX*. <https://www.vexrobotics.com/217-9191.html>

Vujovic, V., & Maksimovic, M. (2014). Raspberry Pi as a Wireless Sensor node:

Performances and constraints. *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1013–1018. <https://doi.org/10.1109/MIPRO.2014.6859717>

Wang, X., & Guo, W. (2009). The Design of RS232 and CAN Protocol Converter Based on PIC MCU. *Computer and Information Science*, 2(3), p176.

<https://doi.org/10.5539/cis.v2n3p176>

- Weinstein, R. (2005). RFID: A technical overview and its application to the enterprise. *IT Professional*, 7(3), 27–33. <https://doi.org/10.1109/MITP.2005.69>
- wemos cc. (2019). *LOLIN D1 mini—WEMOS documentation*.  
[https://docs.wemos.cc/en/latest/d1/d1\\_mini.html](https://docs.wemos.cc/en/latest/d1/d1_mini.html)
- Wibowo, S. B., Putra, G. D., & Hantono, B. S. (2014). Development of embedded gateway for Wireless Sensor Network and Internet Protocol interoperability. *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 1–4. <https://doi.org/10.1109/ICITEED.2014.7007920>
- Zhou, K., Taigang Liu, & Lifeng Zhou. (2015). Industry 4.0: Towards future industrial opportunities and challenges. *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, 2147–2152.  
<https://doi.org/10.1109/FSKD.2015.7382284>

## **ANEXOS.**

# Wireless Adapter Module Development for Robot Communication in IoT Ecosystems

Kevin B. Ruiz-López\*, Luis C. Básaca-Preciado\*, Moises J. Castro-Toscano\*, Yamel Ungson-Almeida\*, Verónica Rojas-Mendizabal\*, Arnoldo Díaz-Ramírez‡, and Wendy Flores-Fuentes†

\* Centro de Innovación y Diseño, CETYS Universidad, Mexicali, Baja California, México

†Facultad de Ingeniería, Universidad Autónoma de Baja California, Mexicali, Baja California, México

‡ Department of Computer Systems, Tecnológico Nacional de México/IT de Mexicali, Mexicali, Baja California, México

**Abstract**—We are witnessing a trend to automatize industrial and daily activities. In this scenario, autonomous robots, which include vehicles and UAVs are becoming used more frequently. These systems consist of many embedded devices to monitor and control the vehicle's activities. The shortage of computational resources of the embedded devices used by robots, such as secondary storage or processor power, are compensated with the use of the cloud. However, many of the devices used in autonomous vehicles lack wireless communication capabilities, which makes it difficult to send information to the cloud provider. Therefore, this paper presents a practical solution for electronic devices capabilities' upgrade by the development of a wireless robot adapter module, allowing communication between devices with wired and wireless protocols. The developed module was tested in an autonomous vehicle. The results of its evaluation are discussed.

**Index Terms**—Wired Communication Protocols, Cloud Services, Adapter Module, Robot Communication, Autonomous Vehicle, IoT.

## I. INTRODUCTION

Nowadays, most of the applications used in daily life, in the industry, and in other areas, are connected to the Internet and are intercommunicated with other applications or devices, especially in the case of robots applied to Internet of Things (IoT) and Industry 4.0 (I4.0). As a consequence, these applications are able to integrate different functionalities, such as remote monitoring, fast access to information, cloud processing and even decision-making for autonomous systems [1]. However, what will happen to devices that do not have a module to access the Internet? Are they destined to be obsolete due to the fact of not being compatible with these new technologies? Wireless Sensor Network (WSN), a core technology for IoT, allows short-range, energy efficient wireless communication with other devices [2], [3]. Over the last few years, researchers have used WSNs in IoT applications [4], [5].

The modern industry is tending towards its fourth transformation [6], [7], where devices and robots inside of the industry can be connected to the Internet to create cyber-physics ecosystems, which will be a relevant and important fact to keep in mind for the future of many companies. Thus, new technologies and protocols are being developed for robot communication to achieve these changes in the industry. In particular, it will be necessary to innovate the process and the equipment that is used in these ecosystems.

In a previous work [8] a first design approach of an autonomous vehicle was presented with the purpose of offering new services to the academic community, such as transportation of community members through the University Campus. In this first prototype the main electronic devices required wired communications to connect to a central computer, showing limitations that directly affect its autonomy.

Therefore, this work presents a solution to the problem of connectivity for communication devices that lack a wireless module to access the Internet. In addition to a methodology that after being applied to an autonomous vehicle, results in the development of a wireless robot adapter module. The module has the purpose to update and improve the capabilities of the autonomous vehicle, by developing a communications module. This will enable interoperability between devices that do not have Internet access through the use of a microcontroller [9], specifically devices that have wired protocols over wireless, such as UART, I2C, CAN, SPI, among others.

The main contribution of this paper is to introduce an adapter module to communicate different kinds of digital devices such as microcontrollers, microcomputers and sensors for robotics applications that use wired communication protocols. The applied methodology allows the selection of an adapter with wireless communication capabilities, to integrate it with cloud services in order to store and process the data acquired by the devices. The mentioned data is used for the operation of an autonomous vehicle, which sends it to a server to be accessed by a website or an IoT app. For instance, consider the case of an autonomous vehicle navigating around the campus most of the time, where some devices and controllers do not have wireless communication capabilities. Therefore, the proposed solution allows selecting a device that uses a wireless protocol (IEEE 802.11 or Wi-Fi), at the same time is compatible with a high diversity of wired communication protocols, which allows the connection of different devices at the same time in order to make them cloud accessible, regardless of the software and hardware platform that the device uses. Additionally, this Wi-Fi programmable device could be connected to any wired sensor directly and work as a gateway to communicate with other devices in the IoT environment with protocols such as MQTT or HTTP and connect to local servers such as Amazon Web Services (AWS) and Google Cloud.

The related work, including WSN architecture and the connection with IoT to improve the communication between robots are briefly explained in Section 2. Some characteristics from wired protocols such as the quantity of devices connected to the bus, frequencies, among others are described in Section 3. The test and results are presented in Section 4 with the diagram of the connections and software used. In section 5 the conclusions and future work are discussed.

## II. RELATED WORKS

Different devices have been developed to allow communication between machines, robots and even autonomous vehicles [10]–[15]. Initially, WSN’s purpose was to communicate devices within a short range [2], [3]. While WSN evolved towards IoT, many protocols of wireless communication, such as IEEE 802.11, RF, Bluetooth, IR, ZigBee among others, have been improved and standardized [16]. However, IoT requires WSN environments to include the use of communication protocols to connect IoT applications to the cloud [4], [5]. Nowadays, permanent communication between devices has been sought wirelessly, regardless of the area of application. For instance, a newer development is swarm robot communication [10], where different types of sensors in separate robots needed to be communicated to carry out more complex tasks and reduce time of operations.

IoT’s added functionality over WSNs, consists of having a network of sensors and devices, connected through the Internet with better coverage area. To achieve this purpose, researchers developed devices that worked as IoT gateways for these sensor networks that were already established [9], [16]–[18] and developed models such as Edge and Fog computing, to efficiently integrate IoT applications and the cloud. The next step is to search for new and better communication protocols for IoT with better response times and lower power consumption [19]–[22].

In reference to our previous work [8], an autonomous vehicle project is presented. Its purpose is to transport people in a fast and safe way inside a university campus, between parking lots, buildings and other specific areas of interest. This vehicle uses several sensors to map the environment and generate information to plan the trajectory for the vehicle, using inertial sensors and strapdown navigation methods [23]. It also incorporates a control and navigation system to send signals to motors and other actuators [24]. The main internal components of the autonomous vehicle development include a NI roboRIO Advanced Robotics Controller, which possess a dual-core ARM real-time Cortex-A9 processor and a customizable Xilinx FPGA, an ATmega328 microcontroller, an ATmega2560 microcontroller, a Raspberry Pi 3 microcomputer, a Wemo D1 mini microcontroller, and a Victor SPX motor driver, among others. These components require both internal and external communication, thus the need for a wireless robot adapter module.

## III. METHODOLOGY

The applied methodology is designed to allow multiple devices, using wired communication protocols to access the Internet through the implementation of a wireless communication module. The purpose of adding wireless robot communication in the autonomous vehicle is to be able to remotely monitor the vehicle’s sensors’ status and to improve its communication capabilities. See Fig. 1.

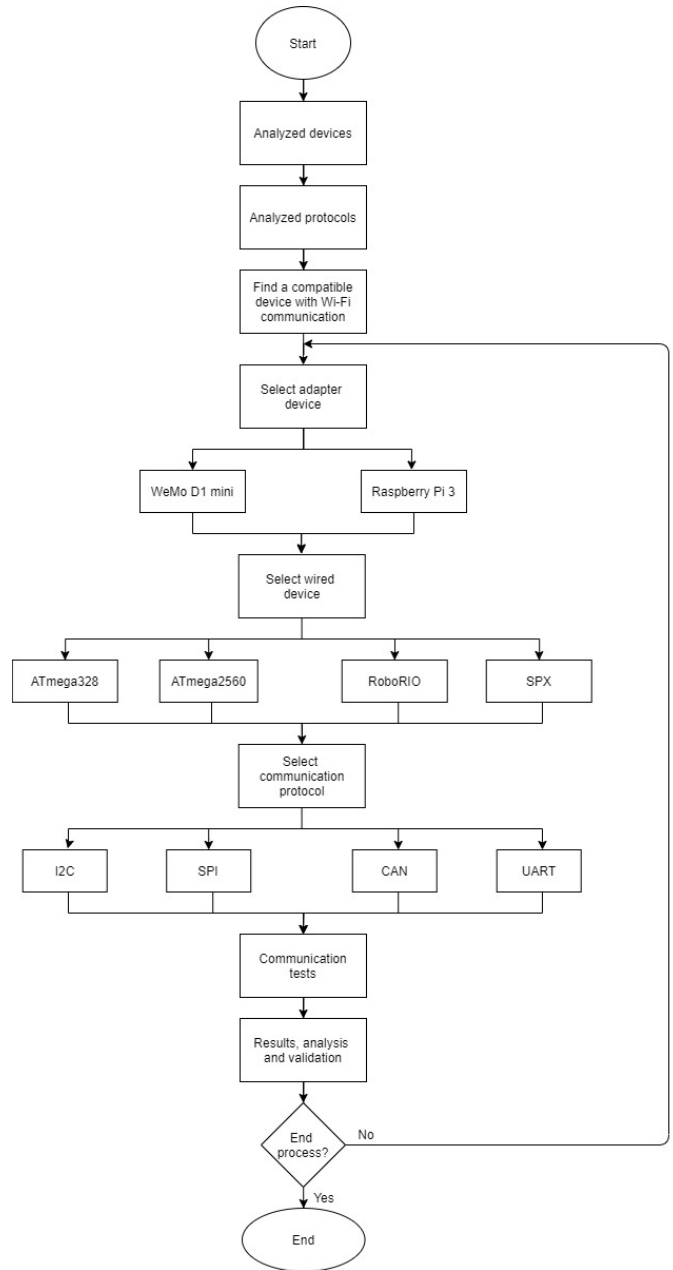


Fig. 1. Proposed methodology based on multiple communications protocol.

The first step consists in reviewing the devices which integrate the autonomous vehicle’s control, power and navigation systems, following with the analysis of its communication protocols, as well as the characteristics of frequencies and

data transfers. The analyzed devices were the controllers and sensors mentioned in the last section. The analyzed protocols are I2C, SPI, UART and CAN. Some considered constraints were that at least two different devices share the same protocol and have at least one line of communication.

According to the aim of this work, it is required to find a compatible microcontroller with all the mentioned protocols and Wi-Fi communication, which allows the integration of different devices.

The following step is to select a wireless communication module. Then, we have to select the wired device to be adapted with Wi-Fi communication. Followed by the selection of a protocol compatible with both devices which will be subject to evaluation by a communication test. After that, it is needed to analyze the test's results and validate communication compatibility. The last step of this methodology is the choice to end the test after every possibility of interaction was tested.

In other words, a configuration procedure between the microcontroller and the devices must be done. Once the final integrations are done, some tests demonstrate the performance of the adapter module.

#### IV. TEST AND RESULTS

Experimentation is conducted to verify the results of the communication of different digital devices, such as microcontrollers, microcomputers and sensors for the autonomous vehicle [8]. Next, a brief description of the tasks of each device used in the control system of the autonomous vehicle before applying the proposed methodology.

The advanced robotics controller RoboRIO, from National Instruments, is used to read the encoder sensors. The next device is the Raspberry Pi, which is used for manual mode where the vehicle is driven with a controller that guides direction and velocity. The ATmega2560 microcontroller is used to read data from the ultrasonic sensors installed around the vehicle to monitor pedestrians or obstacles near the vehicle's surrounding area. These sensors are also used to send an emergency stop when certain conditions are met. ATmega328 microcontroller is used to read messages from the Raspberry Pi and then convert them to PWM signals, these signals will be then sent to motor drives (Victor SPX) later to control the motors of all wheels in autonomous mode. Lastly, as a secondary measure, the WeMo D1 mini was implemented to read all the encoders sensors on each wheel of the vehicle and finally they communicated with the ATmega328 microcontroller to centralize velocity data.

The characteristics of the protocols that will be encompassed are the maximum number of devices interconnected simultaneously, the number of bytes that can be transferred in one interaction, whether the communication is master-slave, on a bus, or point to point, which is an advantage of the protocol.

I2C (Inter-Integrated Circuit), is a multi-master communication protocol, where 128 devices will be able to be connected at the same time, and because this protocol uses 7 bits of the address, it would be able to send n bytes for interaction. I2C can run at three different speeds: standard (100kb/s), fast

(400kb/s), and high speed (3.4Mb/s). Its main feature is that only two signal lines are required to execute this protocol (SDA and SCL) [25].

SPI (Serial Peripheral Interface), is a single-master communication protocol, where the quantitative directly depends on the Slave Select (SS) signal that the master has. It does not have a standard frequency nor a specific addressing outline. This protocol only allows the transfer of one byte for interaction. Additionally, SPI protocol allows straightforward communication with the slave devices [25].

UART (Universal Asynchronous Receiver and Transceiver), is a point to point communication protocol. Therefore only two devices can be connected to communicate with each other. Two signal lines are needed (TXD and RXD) to send n bytes of data between the devices. The most standard baud rate is 9600, but it could be higher depending on the clocks of the devices. However, they need to be the same in both devices. Its advantage is the asynchrony to communicate with other devices [26].

CAN (Controller Area Network), is a multi-master communication protocol. In the bus, up to 110 devices can be connected appropriately. In this protocol, the frequency depends on the distance between devices; i.e., if one component is closer to the other, the frequency would be higher. For real-time tasks the CAN protocol is the most widely used due to not requiring special needs for the transmission medium [27].

Table I shows the communication protocols used by the analyzed devices: NI roboRIO, ATmega328 microcontroller, ATmega2560 microcontroller, Raspberry Pi 3, WeMo D1 mini and Victor SPX. To start with the test, Table 1 shows what protocols each device has out of the box.

TABLE I  
COMMUNICATION PROTOCOLS OF THE ANALYZED DEVICE.

	NI	ATmega		Raspberry Pi	WeMo	Victor
	roboRIO	328	2560	3	D1 mini	SPX
I2C	Yes	Yes	Yes	Yes	Yes	No
SPI	Yes	Yes	Yes	Yes	Yes	No
UART	Yes	Yes	Yes	Yes	Yes	No
CAN	Yes	No	No	No	No	Yes

To determine the device from which the Wireless Robot Adapter Module would be based on, a Raspberry Pi microcomputer and a WeMo D1 mini microcontroller were considered since both have Wi-Fi communication. This allows the wired devices to connect to the Internet in order to access a database and display information in a dashboard to be monitored by the user.

After analyzing the existing devices in the autonomous vehicle's control system, four devices (NI roboRIO, ATmega328 microcontroller, ATmega2560 microcontroller and Victor SPX.) were determined as essential and were selected to be tested for data transfer. The experiment consisted in determining the compatible protocols between the selected devices according to the methodology and a half-duplex data transfer with validation, accuracy and repeatability tests

TABLE II  
PROTOCOLS RESPONSE AVERAGE TEST ON THE ANALYZED DEVICES.

Protocols	Average Response Time (s)	Tests Performed	Average Data Accuracy (%)	Successful Connections (%)	Data Type
SPI	0.000054701699	1000	96.30%	100.00%	Byte (Int)
I2C	0.000517495327	1000	100.00%	100.00%	Byte (Int)
UART	0.007411107	1000	99.90%	100.00%	String

The issues detected in each communication protocol will be discussed. Despite the capabilities of I2C, it generated noise when long arrays of data were sent to the WeMo D1 mini, whereby it generated junk information which was not acceptable for this application. According to the results presented on Table II, the I2C protocol in the Raspberry Pi functioned with an average response time of  $54.7\mu s$  being the second fastest of the tested protocols, however, I2C presented a full percentage of communication-transfer data ratio, compared to the other protocols. The next one is SPI. The quantity required of connection to Slave Select (SS) that the project needs is at least three SS signals, however the WeMo only has one SS and the Raspberry Pi has two SS, making it an issue of limited connections. The CAN protocol does not present issues; however, it's only compatible with two devices (NI roboRIO, Victor SPX), where none has embedded Wi-Fi module, so it would be required to add an external module. See Table I.

The selected option was the UART protocol. The reasons to select this protocol are:

- It presented the most consistency and stability between the device's communication.
- It presented a very high accuracy of data transfer.
- No false-positives or junk information were detected.

Even though the WeMo D1 mini could manage a single encoder, it was not able to communicate with the other three devices at the same time due to the fact that it only has one UART port. To solve this issue, the ATmega2560 was used as a hub, behind the WeMo D1 mini to receive data from the NI roboRIO, the ATmega328 and sensors that are directly connected to it and then send the data to the WeMo D1 mini. The ATmega328 was kept with the purpose of reading the ultrasonic sensors and to send PWM signals to the motor drives. See Fig. 2

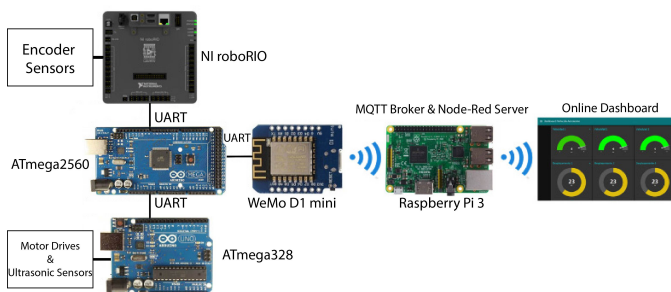


Fig. 2. Final Devices Architecture.

Therefore, the result of applying the developed methodology

was the selection of the WeMo D1 mini in addition with the ATmega2560 microcontroller. The collaborative work of these two devices is intended to be called Wireless Robot Adapter Module. This resulting module was used to communicate the internal devices of the autonomous vehicle with the Internet to make them cloud accessible. See Fig.3.

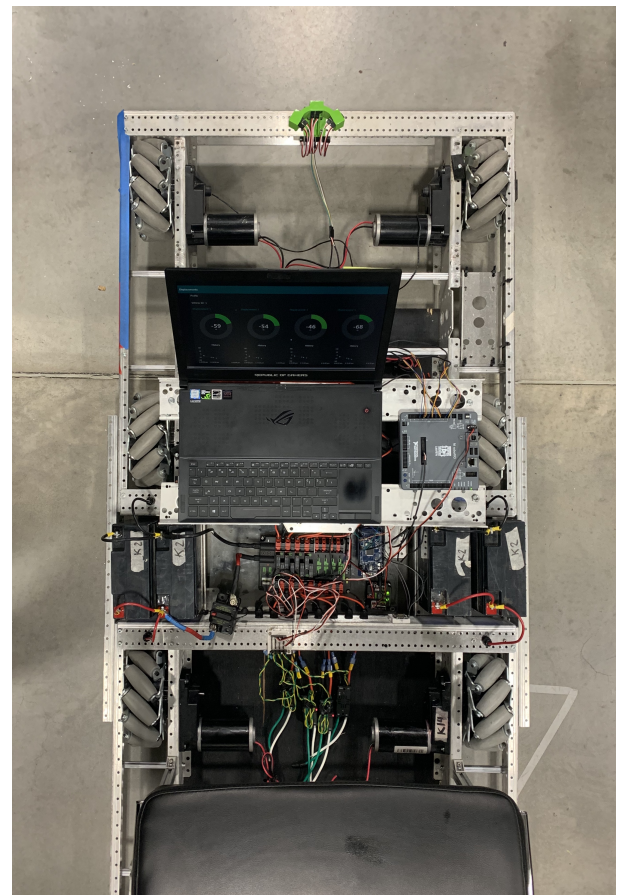


Fig. 3. Autonomous vehicle prototype upgraded with Wireless Robot Adapter Module .

Raspberry Pi is being used to work as an application server to store and to display data in a dashboard. Hence, at the same time, it was also being used as a broker for the IoT protocol that was selected, which is MQTT.

The mentioned application server with the raspberry Pi was implemented using Node-RED, due to its advantages to reach fast implementation such as its visual programming language, as shown on Fig. 4. Node-RED [28] is an open source flow-based programming tool with a browser-based editor capable



of full deployment, adding new libraries from the community, displaying information on websites, among other functions.

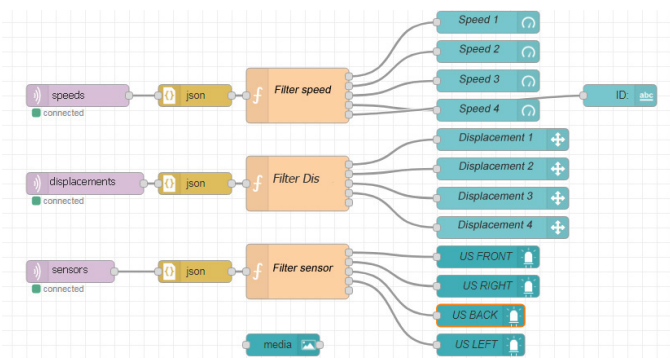


Fig. 4. Node Red Visual Flow-Based Programming.

As a result of the integration of the Node-RED server and the MQTT broker into the Raspberry Pi, it was possible to obtain data from the autonomous vehicle into cloud services in real time. Fig.5 and Fig.6 show the designed dashboards of displacement and speed (respectively) of the autonomous vehicle. These dashboards show the current value and also a history of past values.

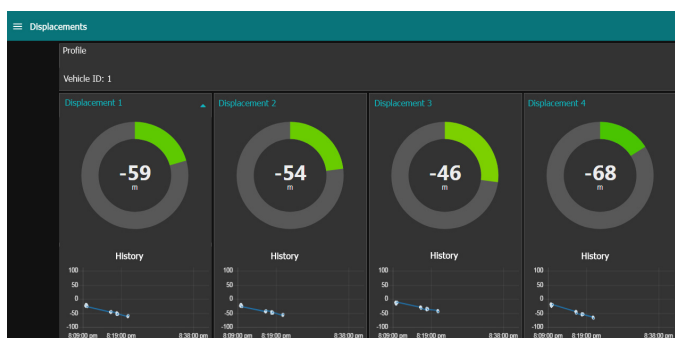


Fig. 5. Dashboard 1 - Displacement.

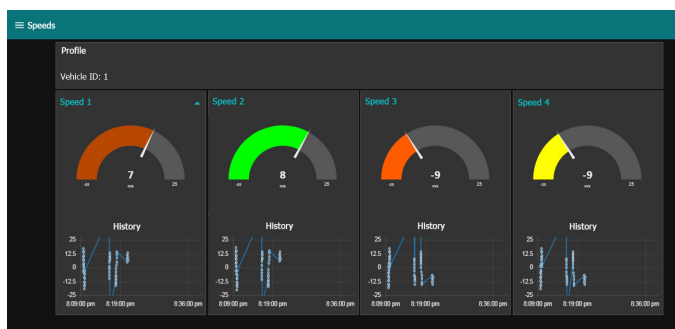


Fig. 6. Dashboard 2 - Speed.

As mentioned before ultrasonic sensors were installed around the vehicle to monitor for pedestrians or obstacles near the vehicle's surrounding area. One set of sensors is located in the front, rear and sides of the vehicle to trigger an emergency stop, to avoid collisions and ensure the safety of the vehicle

users and pedestrians. Fig. 7 shows the dashboard designed to monitor the ultrasonic sensor signals.

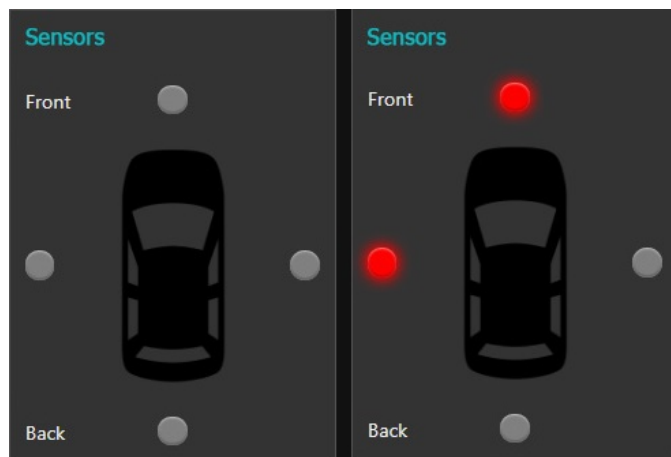


Fig. 7. Dashboard 3 - Sensor

## V. CONCLUSIONS AND FUTURE WORK

The established purpose of this work was achieved. This consisted in allowing the communication of different kinds of electronic devices (for robotics applications using wired protocols) with the developed Wireless Robot Adapter Module.

The developed methodology was applied to the autonomous vehicle project [8] and the result was the creation of a Wireless Robot Adapter Module composed of a WeMo D1 mini and an ATmega2560, which after implementation and testing proved to be accurate, stable and reliable. Also, the use of this adapter reduced the number of interconnected devices in the system, as well as communication issues, response time and power consumption.

The module allowed communication between devices without wireless communication and devices that do, i.e. providing the non-wireless devices with Internet access. This enabled the acquisition of data from the vehicle to be accessible through an online dashboard.

The next stage of this project is to add the LiDAR sensor the ability to communicate wirelessly to the cloud in order to process and analyze the information outside the vehicle in real time to lessen the necessary computing capabilities on board of the vehicle thus making it lower cost and power efficient.

Another application of this project is to utilize this module adapter in the industry with the purpose of enabling I4.0 implementations by providing machines, robots, sensors, actuators and other industry devices, the ability to communicate between them and cloud services, for monitoring, improving production, identifying problems, performing remote testing among other functions.

## ACKNOWLEDGMENT

This research is supported by Centro de Innovación y Diseño (CEID) and Coordinación Institucional de Investigación de CETYS Universidad and the resources of Consejo Nacional de Ciencia y Tecnología CONACYT, México.

## REFERENCES

- [1] A. Nordrum, "Popular internet of things forecast of 50 billion devices by 2020 is outdated (2016)," *Dosegljivo: https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is-outdated*. [Dostopano: 11. 8. 2017], 2017.
- [2] L. Ferrigno, V. Paciello, and A. Pietrosanto, "Experimental characterization of synchronization protocols for instrument wireless interface," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 3, pp. 1037–1046, 2010.
- [3] V. Vujović and M. Maksimović, "Raspberry pi as a wireless sensor node: Performances and constraints," in *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2014, pp. 1013–1018.
- [4] Q. Chi, H. Yan, C. Zhang, P. Li, M. Mukherjee, and L. Da Xu, "A reconfigurable smart sensor interface for industrial wsn in iot environment," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1417–1425, 2014.
- [5] C. Kruger, A. Abu-Mahfouz, and G. Hancke, "Rapid prototyping of a wireless sensor network gateway for the internet of things using off-the-shelf components," in *2015 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2015, pp. 1926–1931.
- [6] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart factory of industry 4.0: Key technologies, application case, and challenges," *IEEE Access*, vol. 6, pp. 6505–6519, 2017.
- [7] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. IEEE, 2015, pp. 2147–2152.
- [8] L. C. Básaca-Preciado, N. A. Orozco-García, J. M. Terrazas-Gaynor, A. S. Moreno-Partida, O. A. Rosete-Beas, J. Rizzo-Aguirre, L. F. Martínez-Grijalva, and M. A. Ponce-Camacho, "Intelligent transportation scheme for autonomous vehicle in smart campus," in *IECON 2018-44th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2018, pp. 3193–3199.
- [9] D. C. Y. Vargas and C. E. P. Salvador, "Smart iot gateway for heterogeneous devices interoperability," *IEEE Latin America Transactions*, vol. 14, no. 8, pp. 3900–3906, 2016.
- [10] D. A. Patil, M. Y. Upadhye, F. Kazi, and N. Singh, "Multi robot communication and target tracking system with controller design and implementation of swarm robot using arduino," in *2015 International Conference on Industrial Instrumentation and Control (ICIC)*. IEEE, 2015, pp. 412–416.
- [11] T. D. Ngo *et al.*, "morebots: System development and integration of an educational and entertainment modular robot," in *2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS)*. IEEE, 2017, pp. 74–80.
- [12] —, "A distributed communication protocol for modular robotic systems," in *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE, 2016, pp. 1097–1102.
- [13] L. Hu, J. Eichinger, M. Dillinger, M. Botsov, and D. Gozalvez, "Unified device-to-device communications for low-latency and high reliable vehicle-to-x services," in *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*. IEEE, 2016, pp. 1–7.
- [14] A. Z. Alkar and U. Buhur, "An internet based wireless home automation system for multifunctional devices," *IEEE transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1169–1174, 2005.
- [15] A. Savvides and M. B. Srivastava, "A distributed computation platform for wireless embedded sensing," in *Proceedings. IEEE International Conference on Computer Design: VLSI in Computers and Processors*. IEEE, 2002, pp. 220–225.
- [16] S. B. Wibowo, G. D. Putra, and B. S. Hantono, "Development of embedded gateway for wireless sensor network and internet protocol interoperability," in *2014 6th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE, 2014, pp. 1–4.
- [17] O. Bello and S. Zeadally, "Intelligent device-to-device communication in the internet of things," *IEEE Systems Journal*, vol. 10, no. 3, pp. 1172–1182, 2014.
- [18] O. Bello, S. Zeadally, and M. Badra, "Network layer inter-operation of device-to-device communication technologies in internet of things (iot)," *Ad Hoc Networks*, vol. 57, pp. 52–62, 2017.
- [19] G. Durante, W. Beccaro, and H. E. Peres, "Iot protocols comparison for wireless sensors network applied to marine environment acoustic monitoring," *IEEE Latin America Transactions*, vol. 16, no. 11, pp. 2673–2679, 2018.
- [20] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 103–112, 2015.
- [21] N. Yamanaka, S. Okamoto, M. Hirono, Y. Imakiire, W. Muro, T. Sato, E. Oki, A. Fumagalli, and M. Veeraraghavan, "Application-triggered automatic distributed cloud/network resource coordination by optically networked inter/intra data center," *Journal of Optical Communications and Networking*, vol. 10, no. 7, pp. B15–B24, 2018.
- [22] P. Desai, A. Sheth, and P. Anantharam, "Semantic gateway as a service architecture for iot interoperability," in *2015 IEEE International Conference on Mobile Services*. IEEE, 2015, pp. 313–319.
- [23] M. J. Castro-Toscano, J. C. Rodríguez-Quinonez, D. Hernández-Balbuena, M. Rivas-Lopez, O. Sergiyenko, and W. Flores-Fuentes, "Obtención de trayectorias empleando el marco strapdown ins/kf: Propuesta metodológica," *Revista Iberoamericana de Automática e Informática industrial*, vol. 15, no. 4, pp. 391–403, 2018.
- [24] L. C. Básaca-Preciado, O. Y. Sergiyenko, J. C. Rodríguez-Quinonez, X. García, V. V. Tyrsa, M. Rivas-Lopez, D. Hernandez-Balbuena, P. Mercorelli, M. Podrygalo, A. Gurko *et al.*, "Optical 3d laser measurement system for navigation of autonomous mobile robot," *Optics and Lasers in Engineering*, vol. 54, pp. 159–169, 2014.
- [25] F. Leens, "An introduction to i2c and spi protocols," *IEEE Instrumentation & Measurement Magazine*, vol. 12, no. 1, pp. 8–13, 2009.
- [26] Y.-y. Fang and X.-j. Chen, "Design and simulation of uart serial communication module based on vhdl," in *2011 3rd International Workshop on Intelligent Systems and Applications*. IEEE, 2011, pp. 1–4.
- [27] X. Wang and W. Guo, "The design of rs232 and can protocol converter based on pic mcu," *Computer and Information Science*, vol. 2, no. 3, pp. 176–181, 2009.
- [28] A. Node-Red, "visual tool for wiring the internet-of-things," Available: <http://nodered.org> (accessed 10th January, 2016), 2016.

Oleg Sergiyenko  
Wendy Flores-Fuentes  
Paolo Mercorelli *Editors*

# Machine Vision and Navigation



Springer

# Machine Vision and Navigation

Oleg Sergiyenko • Wendy Flores-Fuentes  
Paolo Mercorelli  
Editors

# Machine Vision and Navigation

 Springer

*Editors*

Oleg Sergiyenko  
Universidad Autónoma de Baja California  
Mexicali  
Baja California, Mexico

Wendy Flores-Fuentes  
Universidad Autónoma de Baja California  
Mexicali  
Baja California, Mexico

Paolo Mercorelli  
Leuphana University of Lueneburg  
Lueneburg, Low Saxony  
Niedersachsen, Germany

ISBN 978-3-030-22586-5                      ISBN 978-3-030-22587-2 (eBook)  
<https://doi.org/10.1007/978-3-030-22587-2>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

# Chapter 15

## Autonomous Mobile Vehicle System Overview for Wheeled Ground Applications



**Luis Carlos Básaca-Preciado, Néstor Aarón Orozco-García,  
Oscar A. Rosete-Beas, Miguel A. Ponce-Camacho, Kevin B. Ruiz-López,  
Verónica A. Rojas-Mendizabal, Cristobal Capiz-Gómez,  
Julio Francisco Hurtado-Campa, and Juan Manuel Terrazas-Gaynor**

### Acronyms

2D	Two dimensional
3D	Three dimensional
AS/RS	Automated storage and retrieval system
BDS	BeiDou Navigation Satellite System
CML	Concurrent mapping and localization
CPR	Cycles per revolution
DGPS	Differential global positioning system
DoF	Degree of freedom
FMCW	Frequency-modulated continuous wave
FPGA	Field programmable gate array
GNSS	Global navigation satellite system
GPS	Global positioning system
IR	Infrared radiation
IRNSS	Indian Regional Navigation Satellite System
IT	Information technologies
IMU	Inertial measurement unit
LiDAR	Light detection and ranging
MAV	Micro aerial vehicle
MEO	Medium earth orbit

---

L. C. Básaca-Preciado (✉) · N. A. Orozco-García · O. A. Rosete-Beas · M. A. Ponce-Camacho  
K. B. Ruiz-López · V. A. Rojas-Mendizabal · C. Capiz-Gómez · J. F. Hurtado-Campa  
J. M. Terrazas-Gaynor  
CETYS Universidad, Mexicali, Mexico  
e-mail: [luis.basaca@cetys.mx](mailto:luis.basaca@cetys.mx); [nestor.orozco@cetys.mx](mailto:nestor.orozco@cetys.mx); [oscar.rosete@cetys.mx](mailto:oscar.rosete@cetys.mx);  
[miguel.ponce@cetys.mx](mailto:miguel.ponce@cetys.mx); [kevinbennett.ruiz@cetys.edu.mx](mailto:kevinbennett.ruiz@cetys.edu.mx); [veronica.rojas@cetys.mx](mailto:veronica.rojas@cetys.mx);  
[cristobal.capiz@cetys.mx](mailto:cristobal.capiz@cetys.mx); [juan.terrazas@cetys.mx](mailto:juan.terrazas@cetys.mx)

MUTCD	Manual on uniform traffic control devices
OD	Obstacle detection
PPR	Pulses per revolution
RADAR	Radio detection and ranging
ROV	Remotely operated vehicle
SAE	Society of Automotive Engineers
SLAM	Simultaneous localization and mapping
SoC	System on a chip
S/R	Storage and retrieval
ToF	Time of flight
TSR	Traffic sign recognition
UL	Unit load
VO	Visual odometry

## 15.1 Introduction

We are in the era of liquid modernity [1], where the fragility of markets lays on the power of the will of new generations. Post-centennial generation who will be born around 2020 or even already were born will grow up with the speed of 5G, Internet of things, and augmented reality, between other disruptive technologies yet unknown. In the era of innovation, products will be mostly services [2, 3] based on value propositions [4] that fit with the profile of a market that will use disruptive technologies. The use of autonomous electric cars depends on the evolution of business models based on open services. Clearly, with the advent of 5G technology, disruptive technologies will define the future market trends. One of those disruptive technologies will be autonomous electric cars.

Why electric autonomous cars? First off because autonomous cars or self-driving cars have all the features of business trends. Those businesses will be ruled by IT platforms and consumer profiles of the markets. Innovation is changing the future of successful business models. The next generations, after Z generation, will choose to buy or rent products and services online. Business based on IT platforms will ease the use of autonomous electric cars, in an era when big and expensive cars will be obsolete. Perhaps an icon like Amazon which has no factories can teach us the generational path to the new generation practices of individual transport. We understand why people prefer buying online. It is easier, cheaper, faster, and even functional. A single place to pick up from a variety of choices.

Individual transport as a service will change in the next 15 years, because of the natural timeline of future drivers. According to H. Chesbrough, the next generations of entrepreneurs will embrace new paradigms to develop businesses. So the commitment to the future of cars is a combination of two generations, centennials and post-centennials, and the use of business models based on IT platforms. The innovative business committed to fit customer needs that use to buy or rent online but with a power one thousand times faster than their previous generation because of 5G technologies. In the near future, innovation on products as a service will be a practice on a regular basis, so autonomous electric cars are in the center of the



trend. Products as a service are a trend in the business model of a lot of companies. Nowadays more and more, most of the revenues for enterprises come from services instead of products. Of course, technology is important, but the way the value proposition is delivered is more important.

So, at the end of the day, owning a product in the era of liquid modernity will not be relevant, but the service delivery will be. Perhaps disruptive business models in the near future will change the way people transport or get things transported on each day basis, leaving behind the madness of gasoline cars as a heavy, huge, and polluting artifact for museums. That day is not way long far, as well as future drivers will be born around 2020, or even were already born and right now are preparing to ask for a lift from an autonomous electric car through a 5G-IoT graphene device.

Therefore, the following sections present an introduction to what is considered an autonomous vehicle, its main components and applications. Afterward, more in-depth topics are addressed, such as perception (sensors), self-localization, environment mapping, trajectories planning; along with a case study of an electric autonomous vehicle which is currently in development at CETYS University Mexicali Campus in México, for transportation of people inside the campus. And in the end, a perspective on how innovative business models will change the future of cars and transport, along with our conclusions, is provided.

## 15.2 Fundamentals of Autonomous Mobile Vehicles

### 15.2.1 *Levels of Automation*

The current taxonomy and definitions of the automation of an on-road motor vehicle are governed by the J3016 standard, originally published by the Society of Automotive Engineers (SAE International) in 2014 and revised in 2016 and 2018 [5]; they are organized in a classification system consisting of six different levels (see Fig. 15.1):

- **Level 0**, No Automation: In this level of automation, the entire performance of the motor vehicle is under the authority of a human driver, this including all the tasks related to driving, such as the execution of steering, acceleration/deceleration maneuvers, monitoring and sensing the driving environment, and the fallback performance of the dynamic driving task, which is the operational and tactical aspects of driving a car, but not the strategic ones, such as choosing a destination. Most commercial motorized vehicles fall under this category of being driven fully by humans.
- **Level 1**, Driver Assistance: in this level of automation, the entire performance of the motor vehicle still falls under the authority of a human driver, but under certain circumstances, and using information of the driving environment, an integrated assistance system can aid the human driver in certain tasks such as steering, or accelerating/decelerating, all under the assumption the driver will take care of the remaining aspects of the drive. It must be noted that a motor vehicle may have



## SAE J3016™ LEVELS OF DRIVING AUTOMATION

	SAE LEVEL 0	SAE LEVEL 1	SAE LEVEL 2	SAE LEVEL 3	SAE LEVEL 4	SAE LEVEL 5
What does the human in the driver's seat have to do?	You are driving whenever these driver support features are engaged – even if your feet are off the pedals and you are not steering			You are not driving when these automated driving features are engaged – even if you are seated in "the driver's seat"		
	You must constantly supervise these support features; you must steer, brake or accelerate as needed to maintain safety			When the feature requests, you must drive	These automated driving features will not require you to take over driving	
What do these features do?	These are driver support features			These are automated driving features		
	These features are limited to providing warnings and momentary assistance	These features provide steering OR brake/acceleration support to the driver	These features provide steering AND brake/acceleration support to the driver	These features can drive the vehicle under limited conditions and will not operate unless all required conditions are met	This feature can drive the vehicle under all conditions	
Example Features	<ul style="list-style-type: none"> <li>• automatic emergency braking</li> <li>• blind spot warning</li> <li>• lane departure warning</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering OR</li> <li>• adaptive cruise control</li> </ul>	<ul style="list-style-type: none"> <li>• lane centering AND</li> <li>• adaptive cruise control at the same time</li> </ul>	<ul style="list-style-type: none"> <li>• traffic jam chauffeur</li> </ul>	<ul style="list-style-type: none"> <li>• local driverless taxi</li> <li>• pedals/steering wheel may or may not be installed</li> </ul>	<ul style="list-style-type: none"> <li>• same as level 4, but feature can drive everywhere in all conditions</li> </ul>

Fig. 15.1 Levels of driving automation by SAE [5]

one or more tasks aided by said systems, depending on the number of system capabilities, or driving modes.

- **Level 2, Partial Automation:** In this level of automation, a driving mode executes a specific task of the steering, and acceleration/deceleration ADAS capabilities. Certain parts of the driving experience can be automated in this level, with systems such as self-parking, following a car at a certain distance and staying in lane. The driver is always in control of the car.
- **Level 3, Conditional Automation:** In this level of automation, there is an autonomous driving mode which uses data of the driving environment obtained through an array of advanced sensors, typically ultrasonic, RADARs, LiDARs, and machine vision. These cars can make decisions and perform the same and more advanced tasks than the previous level. Nevertheless, this level is still not fail-safe, and drivers must be ready to take control of the vehicle at all times if the situation requires it.
- **Level 4, High Automation:** In this level of automation, all capabilities of the previous level are available, the difference is that these systems possess more advanced technology, that is, redundant systems which allows the vehicle to handle any situation or system failure on its own. Passengers are not required to supervise but may take control of the vehicle if desired or preferred. These features are limited only to certain areas or conditions and if the required conditions are not met, autonomous features will be limited or will not engage.
- **Level 5, Full Automation:** In this level of automation (which is the highest), similarly to level 4, passengers do not need to supervise and the vehicle can

perform full autonomous driving in any conditions while adapting to any driving situation and guaranteeing the safety of passengers, pedestrians, and other drivers. These features are not limited to specific areas or conditions. These vehicles can drive everywhere while offering a more responsive and refined experience for the passenger on board.

Currently, there is no autonomous vehicle that is level 5 but companies such as Google Waymo and Tesla (among others) are working hard to reach this level and it is only a matter of time before they do. See Sect. 15.2.3 for a commercial autonomous vehicle application, the Google Waymo's self-driving taxi.

### 15.2.2 Main Components

The components of an autonomous vehicle vary depending on the application of the vehicle; however, there are certain main components that are indispensable inside a vehicle to be able to achieve a level of autonomy.

Central or main computer: This computer can be from a microcontroller or a Field Programmable Gate Array (FPGA) to a complete computer or a mini computer. It should be mentioned that depending on the design architecture implemented in the vehicle, it is possible to find more than one computer connected and working for the central computer. Additionally, it is responsible for communications between electronic devices, sensor reading, data processing, trajectory calculation, control of vehicle actuators, among other functions.

Some designs of autonomous low-end and low-cost vehicles that are in development can get to use mini or micro computers such as the raspberry pi [6] or the zini [7]. However most of the designs of autonomous vehicles for industry or commercial use full computers on board, with multicore processors and dedicated graphics cards.

The sensors are some of the indispensable and most important components of an autonomous vehicle because the sensors perform the function of measuring physical variables or desired parameters such as angular position, speed, temperature, acceleration, position, distance, among others. The measurement of these parameters and the analysis of the information they provide is called perception and is basically the ability of a vehicle to be aware of itself and what is happening around it, this being the first step to move toward the autonomy.

The most important sensors found in autonomous vehicles are the following:

Odometry sensors that can be optical or magnetic encoders to measure the angular position and speed of the vehicle's wheels, in order to have the ability to calculate the displacement of the vehicle in real time.

Inertial measurement unit (IMU) is a device that can hold up to three sensors of three dimensions each simultaneously, commonly to each dimension of measurement known as degree of freedom (DoF). The sensors contained in an IMU are a three-axis gyroscope, a three-axis accelerometer, and a three-axis magnetometer. The signals

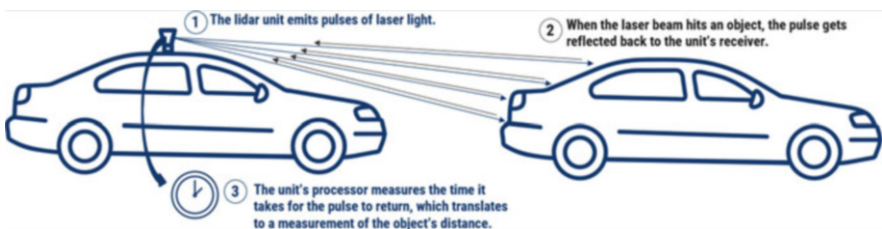
provided by these sensors are used to calculate the angular position and orientation of the vehicle.

Light Detection and Ranging or LiDAR is an advanced sensor used to create digital maps of the surrounding environment of the vehicle, these sensors are complex optoelectronic systems that use moving mechanical parts to control the rotation and speed of 1–128 lasers that are inside the device. The signals of the lasers are emitted and reflected by obstacle surfaces or objects in the field of view of the device; afterward, the reflection is detected and through the technique time of flight or ToF, the distance from the emitting device to the laser reflexion point (on the object surface) is calculated, allowing effectively the use of this information to create digital maps or point clouds of the environment surrounding the LiDAR (see Fig. 15.2). Later, in the perception section of this chapter, the use, advantages, most common models of LiDAR employed in the autonomous vehicle industry, and its characteristics are studied.

The video cameras, similar to the LiDAR, are used to perceive the environment surrounding the vehicle; however, due to the nature of how a video camera operates, the information that can be extracted from the data of the same is very different, and although there are techniques to calculate the depth of some feature of an image, such as stereovision [8], the LiDAR is the most appropriate and fastest to perform the task of creating digital maps of the environment. The main function of a camera is to detect features and patterns within the images captured in video and the identification, location, and classification of objects or figures of interest such as people, other vehicles, and dynamic objects, the lines that divide the lanes in a street or highway, the traffic signals, or landmarks used by the vehicle for the correction of its position, among others.

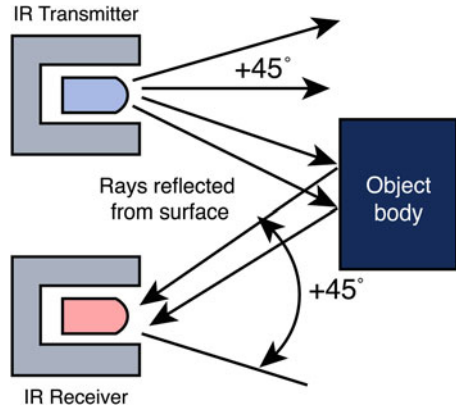
Ultrasonic sensors are mainly used to measure distances or detect the presence of objects within a range of vision. These sensors use a transmitter and an ultrasonic wave receiver and the ToF principle to calculate the distance of the detected object(s). Its main advantage is its wide field of view, which makes this type of sensor able to detect obstacles in close proximity to the vehicle.

Infrared sensors have various applications, but their main function is to detect the presence of objects at a predefined distance using an emitter or transmitter and the detection of the reflection of light as a signal. In Fig. 15.3 the basic principle of operation is shown.



**Fig. 15.2** Simplified diagram of how automotive lidar works

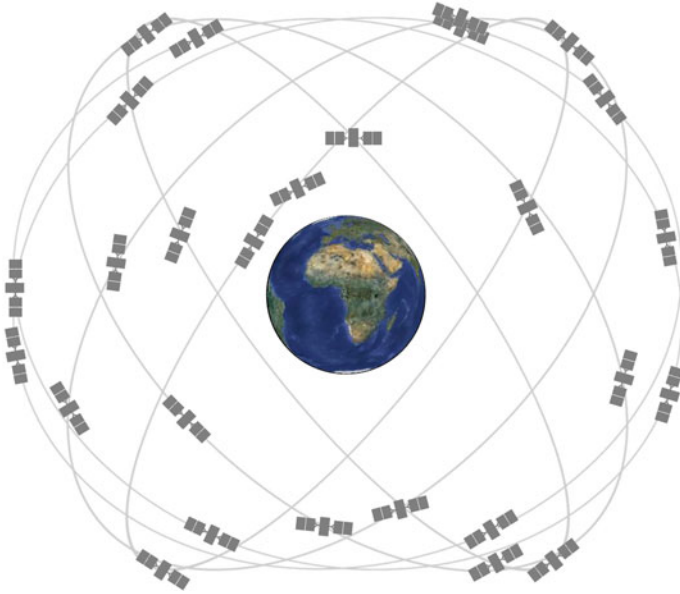
**Fig. 15.3** Infrared sensor basic operation principle



**Fig. 15.4** Satellite GPS III launched in 2018 [9]

As a disadvantage, these sensors have by nature a limited range of vision, unlike ultrasonic sensors. However, they also have a great advantage, and this is their high response speed compared to other sensors such as those based on a principle of resistive transduction. Because of this, these types of sensors are a viable option to quickly detect obstacles in a limited range of vision.

The Global Positioning System or GPS is fundamental in an autonomous vehicle because this system, as its name indicates, is able to calculate its current position in any part of the world, using satellites as a reference (see Fig. 15.4).



**Fig. 15.5** Twenty-four-slot satellite constellation, as defined in the GPS Performance Standard [9]

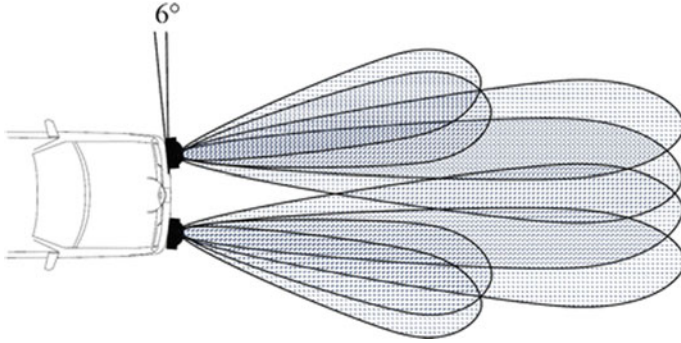
Devices equipped with GPS receive signals from a constellation of satellites which contains at least 24 operational satellites that are flying in medium earth orbit (MEO), at approximately 20,000 km altitude (see Fig. 15.5).

As of January 9, 2019, there were a total of 31 operational satellites in the GPS constellation. The GPS satellite expansion gave as a result improved coverage in most parts of the world [9].

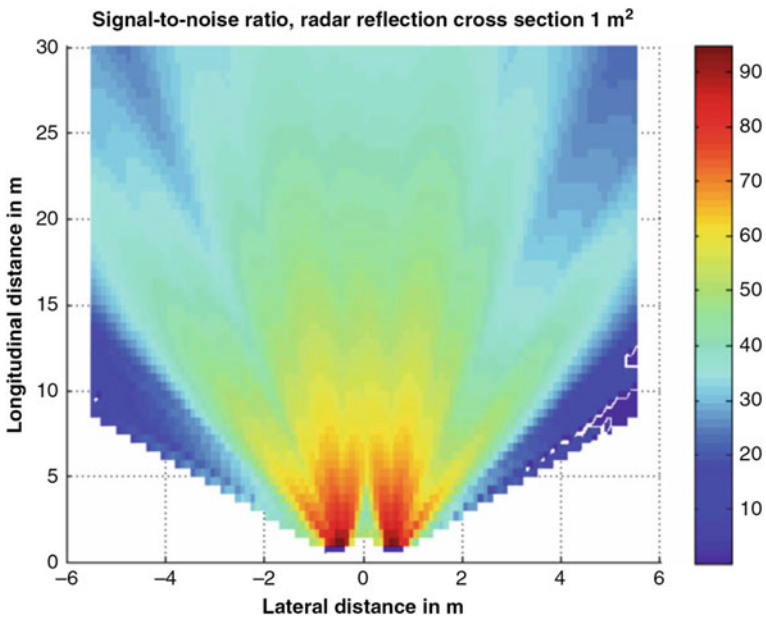
The term GPS is owned by the United States; however, there are other satellite navigation systems which use their own satellite constellation such as BeiDou Navigation Satellite System (BDS) from China [10], the Indian Regional Navigation Satellite System (IRNSS) [11], Galileo from the European Union [12], among others.

Radio Detection and Ranging (RADAR or radar) is a well-known and studied technology developed originally for military applications during the Second World War. In autonomous vehicles, radars are active radio frequency devices used to determine the location, speed, and direction of obstacles or targets of interest within its coverage area [13].

Radars work similar to the LiDAR sensors, the main difference between these systems is that radars use radio waves instead of laser; however, there are advantages and disadvantages between them. LiDARs can detect smaller objects and can build an exact 3D image of an object, while RADARs cannot. On the other hand, RADARs can operate in cloudy weather conditions and possess a longer operating distance, whereas LiDARs have greater difficulty in these conditions [14]. In Fig. 15.6, a dual RADAR configuration for an autonomous vehicle is shown; this configuration combines two RADARs in one, which generates some advantages such as a broader



**Fig. 15.6** Dual RADAR configuration for autonomous vehicle [15]



**Fig. 15.7** Detection coverage of the dual RADAR configuration [15]

coverage, fault detections, and an improved signal processing due to the overlap of both sensors [15] (see Fig. 15.7).

The components mentioned above are the most important or indispensable in any autonomous vehicle development that wants to achieve a high level of autonomy; however, they are not the only ones and new solutions, and technologies for autonomous vehicles are constantly being developed. In the next sections of this chapter, you will find more detailed and in-depth information on the functioning of these components.

### 15.2.3 Applications

Due to the fact that autonomous vehicles are wheeled mobile platforms, they share some of the applications that conventional land vehicles, such as automobiles, tuggers, cranes, and carts have, with the added benefits that automation brings in specific scenarios, such as higher precision, endurance, worker safety, and efficiency, among others. This section will explore common modern autonomous vehicle applications across different industries.

#### 15.2.3.1 Automated Storage and Retrieval System (AS/RS)

Unit load (UL) and storage and retrieval (S/R) systems are a critical link for supply chains in global markets. The introduction of crane and tugger-based technologies allowed for automated vehicle systems to be introduced in UL S/R system configurations, such as the one presented in Fig. 15.8, finding wide implementation in modern storage facilities, built or renovated after 1994, year in which AS/RS systems had a significant increase in implementation in distribution environments in the United States [17]. The main benefits of AS/RS over conventional UL R/S systems are:

- Considerable savings in labor costs.
- Higher efficiency of floor space and storage layout configuration.
- Increased working reliability.
- Reduced error rates and product waste.

Nevertheless, it must be noted that since AS/RS has a higher level of technology integration than conventional UL R/S systems, they tend to be significantly costlier, as a high investment is needed in order to establish the desired level of automation, and a proper closed loop control system; similarly, because of the systematic approach

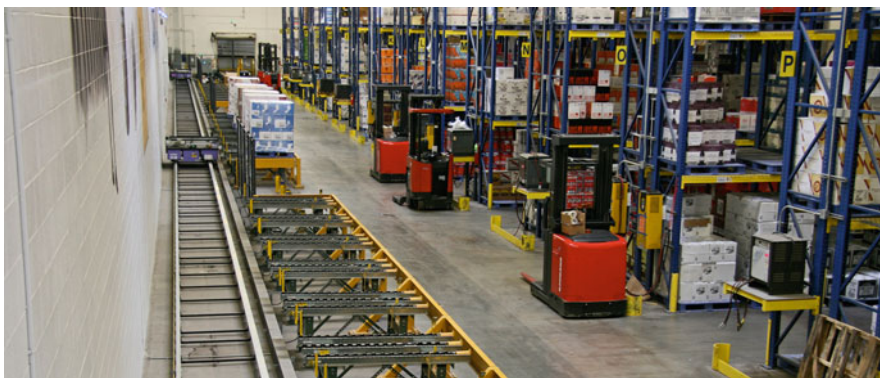


Fig. 15.8 Distribution center with an implemented AS/RS [16]



that automation entails, it tends to be significantly less flexible than conventional, non-automated systems [17, 18].

### 15.2.3.2 Mobile Industrial Robots

Mobile industrial robots are cooperative robots that are specifically programmed to work in an industrial setting (see Fig. 15.9). Their main functionalities are focused on increasing efficiency of industrial processes and low-value logistics tasks, increasing the speed of internal transportation through the use of automation and deep learning, and providing a cost-efficient solution for improving the overall reliability of the industrial environment by using a set of very powerful and precise sensors.

### 15.2.3.3 Commercial Autonomous Vehicles

One of the most common applications of autonomous vehicles, “self-driving cars” are modern solutions that are commercially available with the implementation of automation up to a certain degree. Most commercially available self-driving vehicles fall between the second and third level of automation, with some vehicles in the fourth level currently undergoing trial sessions [19] (see Fig. 15.10). Due to the sensibility of the working environment of these vehicles, namely the fact that human lives are

**Fig. 15.9** MiR500 Mobile Robot for industrial applications [16]



**Fig. 15.10** Google Waymo’s self-driving taxi is in service in Arizona in December 2018 [19]



at stake, the trial period of these systems are considerably more rigorous than for other autonomous land vehicle applications, this with the intention of having greater passenger security, and reducing the amount of disengage incidents, when the driver has to take control of the autonomous system. Due to current level of automation of commercially available self-driving vehicles, it must be noted that, for safety reasons, the driver must be alert during the entire driving task, in case a disengage incident occurs.

#### 15.2.3.4 Autonomous Vacuum Cleaners

Robotic vacuum cleaners, such as the Roomba i7 from iRobot (see Fig. 15.11), have gone from simple closed loop systems to full-fledged deep learning autonomous systems. These autonomous robotic systems use an integrated array of specialized sensors that allow them to navigate through cleaning environments, focusing on avoiding obstacles, such as furniture, while maximizing the efficiency of surface-cleaning through the use of proprietary algorithms. Modern iterations of autonomous vacuum cleaning robots include smart mapping features that automatically generate a layout of the cleaning environment, and labels it accordingly for future cleaning tasks; these systems typically have some degree of IoT integration in the form of an application or through communication with dedicated personal home assistants such as Google Home or Amazon's Alexa.

It must be noted that the applications explored in this section were chosen based on relevance, and notability among the general public. As more levels of automation are achieved, more relevant applications of autonomous systems will emerge; as such, there are other relevant applications currently in development for autonomous land vehicles in sectors such as agriculture, logistics, entertainment, and healthcare.

**Fig. 15.11** iRobot Roomba i7, autonomous robotic vacuum [20]



## 15.3 Perception

### 15.3.1 Environment Sensing

In a general perspective, sensors can be grouped according to the functions they provide. Internal vehicle state sensors provide information about the current operation and state of the vehicle, including lower-level functions such as engine operations and higher-level states such as vehicle motion and position. External environment sensors provide information about the world outside the vehicle, potentially including road and lane information, the location and motion of other vehicles, and stationary physical objects in the world. Finally, driver state and intention sensors provide information about the state or intentions of the driver. These sensors can include seat occupancy and passenger weight (pressure or infrared sensors), audio sensors, internal cameras, eye trackers, breath alcohol sensors, and haptic transducers [21].

There are a number of issues when designing the external environment sensing suite for an autonomous vehicle. The system objectives and requirements are obviously a direct influence in the design process. For example, highway driving is a much more structured environment than off-road driving, and that structure can be exploited in sensor and control system design. On the other hand, the urban environment, which may consist of irregular and changing road networks and vehicles and pedestrians behaving unpredictably, is much less structured than highway driving and may require significantly more sensing capability. Robustness and safe performance of hardware and software are obviously required for both off-road driving and on-road production automobiles. Redundancy in sensing modalities is also a highly desired feature, especially in a less structured, more uncertain environment, but the cost of many of the sensors used on research vehicles would be prohibitive for a commercial passenger vehicle application. The level of autonomy with respect to the human driver is also a significant design issue in a passenger vehicle system. The degree to which the driver is to be part of the sensing and control loop is a design decision driven by both technical and nontechnical (i.e., marketing and legal) considerations.

A number of different sensors have been developed for sensing the external environment of an autonomous vehicle. Many have been developed initially for safety warning or safety augmentation systems that are now being deployed on some high-end vehicles. Among the array of different sensors available for establishing an environment sensing system, the following are worth mentioning:

#### 15.3.1.1 LiDAR Sensor

In principle, a LiDAR consists of a transmitter and a receiver. Short light pulses with lengths of a few to several hundred nanoseconds and specific spectral properties are generated by the laser. Many systems apply a beam expander within the transmitter

unit to reduce the divergence of the light beam before it is sent out into the atmosphere. At the receiver end, a telescope collects the photons backscattered from the atmosphere. It is usually followed by an optical analyzing system which, depending on the application, selects specific wavelengths or polarization states out of the collected light. The selected radiation is directed onto a detector, where the received optical signal is converted into an electrical signal. The intensity of this signal in its dependence on the time elapsed after the transmission of the laser pulse is determined electronically and stored in a computer [22].

LiDARs carry out sensing by using their own emitted light, and therefore, they are not sensitive to environmental illumination. Road detection systems that rely on this type of sensor can then, in principle, provide the same level of accuracy across the full spectrum of light conditions experienced in daily driving; therefore, they are particularly suitable for achieving higher levels of driving automation [23].

Among the existing commercial LiDAR, the following devices are worth mentioning:

- **Velodyne VLP-16 LiDAR**

Capable of delivering the most accurate real-time 3D data that consists of the creation of a full 360° environmental point cloud view (Fig. 15.12).

- **Velodyne HDL-64E**

The HDL-64E LiDAR sensor includes 64 channels with an accuracy of  $\pm 2$  cm and a 100–120 m range, designed for obstacle detection and navigation of autonomous ground vehicles and marine vessels (Fig. 15.13).

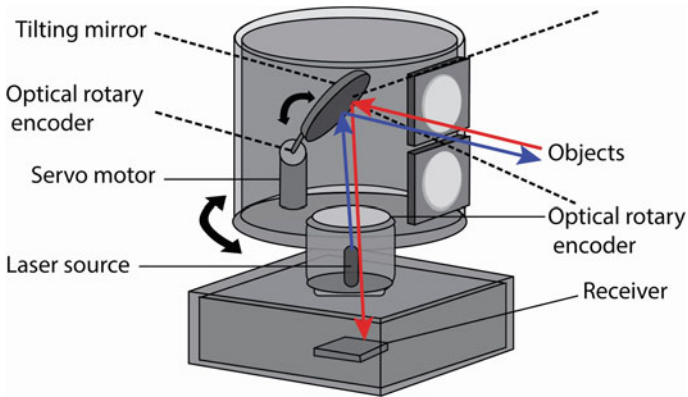
Recently a new LiDAR technology arrived, solid state LiDARs, which have the advantage that they are smaller and, importantly, they eliminate moving parts involved in the optical mechanisms as seen in Fig. 15.14, which may enable their mass manufacture thereby bringing manufacturing costs down [27].

One of the most remarkable solid-state LiDARs InnovizOne with a detection range of 250 m, a horizontal field of view of 120° and a vertical field of view of 25° which comes at a fraction of the cost of the equivalent electromechanical solution (Fig. 15.15).

**Fig. 15.12** Velodyne VLP-16 LiDAR [24]



**Fig. 15.13** Velodyne HDL-64E LiDAR [25]



**Fig. 15.14** Diagram of the LiDAR optics and encoders [26]

**Fig. 15.15** InnovizOne solid-state LiDAR [28]



### 15.3.1.2 Kinect Sensor

The Kinect device is a horizontal bar composed of multiple sensors connected to a base with a motorized pivot. Looking at the Kinect sensor from the front, from the outside, it is possible to identify the infrared radiation (IR) projector, the RGB camera,

**Fig. 15.16** Kinect sensor  
[29]



and the depth camera. An array of four microphones, a three axis accelerometer, and the tilt motor are arranged inside the plastic case (Fig. 15.16).

The device is connected to a PC through a USB 2.0 cable. It needs an external power supply in order to work because USB ports do not provide enough power. The IR projector is the device that Kinect uses for projecting the IR rays that are used for computing the depth data. The IR projector, which from the outside looks like a common camera, is a laser emitter that constantly projects a pattern of structured IR dots at a wavelength around 830 nm [30].

### 15.3.2 *Obstacle Detection and Tracking*

Obstacle detection (OD) is one of the main control system components in autonomous vehicles, since a reliable perception of the real world is a key-feature of any obstacle detection system for dynamic environments. In recent years, most of the historical approaches in literature have been readjusted in the framework of stereo vision and other 3D perception technologies (e.g., LiDAR), and important results have been provided by several experiments on autonomous ground vehicles.

In order to achieve a good performance, most of the OD algorithms take some assumptions about the ground or about the approximated free space on it. Algorithms based on stereo vision and other 2D/3D sensors. Each obstacle detection system is focused on a specific tessellation or clustering strategy; hence, they have been categorized into four main models [31].

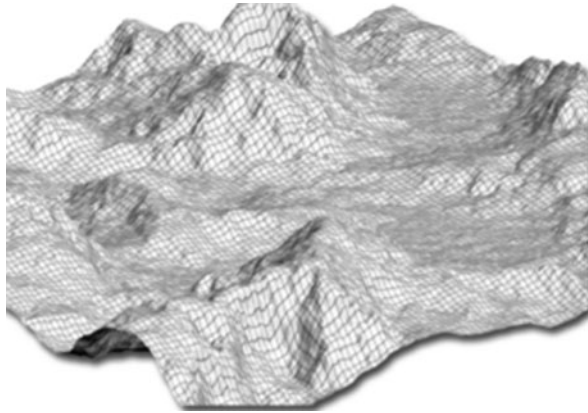
#### 15.3.2.1 **Probabilistic Occupancy Map**

Occupancy grid maps address the problem of generating consistent maps from noisy and uncertain measurement data, under the assumption that the robot pose is known. The basic idea of the occupancy grids is to represent the map as a field of random variables arranged in an evenly spaced grid [32]. As seen in Fig. 15.17, each random variable is binary and corresponds to the occupancy of the location it covers.

**Fig. 15.17** Occupancy grid map obtained from ultrasound data [33]



**Fig. 15.18** Landscape digital elevation map [35]



### 15.3.2.2 Digital Elevation Map

The concept of digital terrain model was defined as a digital (numerical) representation of the terrain. Other alternatives to this term have been used like digital elevation models, digital height models, digital ground models, as well as digital terrain elevation models [34]. Figure 15.18 shows the case of specific digital elevation models which refers to a digital representation of the terrain on a height above a given level, especially that of the sea.

### 15.3.2.3 Scene Flow Segmentation

Scene flow estimation is the challenging problem of calculating geometry and motion at the same time. By considering images from one view point, scene flow estimation is underdetermined. Also, a moving camera still creates ambiguities between camera and scene motion. Only after introducing additional cameras, this situation can be resolved and increase the robustness of the system [36]. Figure 15.19 shows the application of scene segmentation techniques on vehicle detection.

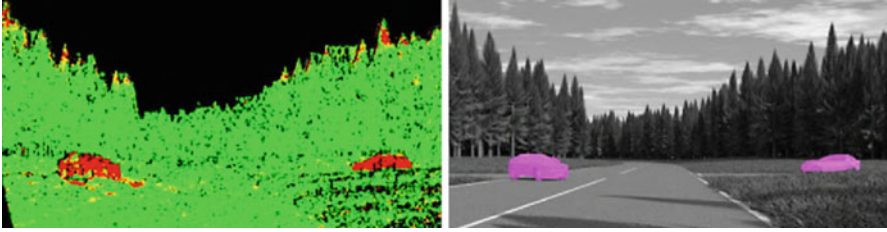


Fig. 15.19 Scene flow segmentation obstacle detection [37]

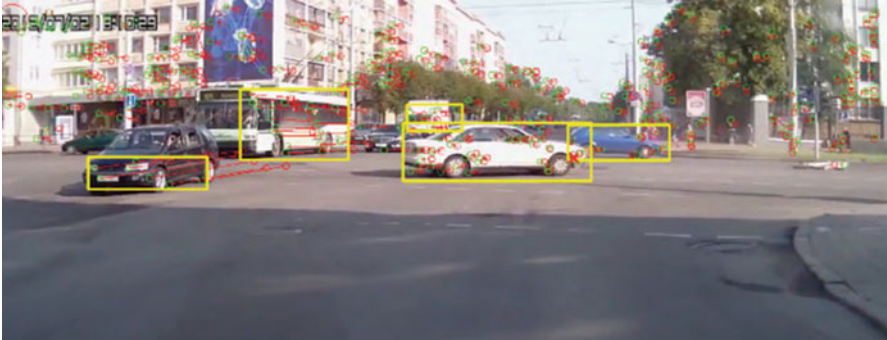


Fig. 15.20 Epipolar geometry-based clustering applied to an urban traffic scene [39]

#### 15.3.2.4 Geometry-Based Clusters

These algorithms are based on a search method that clusters the sensor measurements using a specific geometry model like the double-cone model [38]. As seen in Fig. 15.20, geometry-based clustering techniques can be applied to cities to detect obstacles.

### 15.3.3 Traffic Signs

Traffic signs are markers placed along roads to inform drivers about either road conditions and restrictions or which direction to go. They communicate a wealth of information but are designed to do so efficiently and at a glance. This also means that they are often designed to stand out from their surroundings, making the detection task fairly well defined. The designs of traffic signs are standardized through laws but differ across the world. In Europe, many signs are standardized via the Vienna Convention on Road Signs and Signals. In the United States, traffic signs are regulated by the Manual on Uniform Traffic Control Devices (MUTCD).

While signs are well defined through laws and designed to be easy to spot, there are still plenty of challenges for Traffic Sign Recognition (TSR) systems [40].



1. Signs are similar within or across categories.
2. Signs may have faded or are dirty so they are no longer their original color.
3. The sign post may be bent, and therefore, the sign is no longer orthogonal to the road.
4. Lighting conditions may make color detection unreliable.
5. Low contrast may make shape detection hard.
6. In cluttered urban environments, other objects may look very similar to signs.
7. There may be varying weather conditions.

Among the challenges that must be considered for the adequate detection of traffic signs, lighting, viewpoints, and weather conditions are present. These considerations suggest that relying solely on color is problematic; therefore, shape information is useful for sign detection. An example of the use of shape information is seen in [41].

Traffic sign detection methods include the following [40]:

- Learning-based traffic sign detectors
- Segmentation methods
- Hough transform derivatives

### 15.3.4 Landmarks

Landmarks are features of an environment or scenery that can be easily recognized and thus can enable someone to locate his or her spatial position within a given location. Landmarks in an urban environment tend to be human-made features, including but not limited to:

- Street name signs
- House number plates
- Sidewalk symbology
- Traffic signs
- Billboards
- Buildings
- Monuments
- Other easily recognizable urban features

This is due to the fact that, in urban environments, human-made features tend to be more permanently established and recognizable than nature-based features [42].

Autonomous vehicles often include landmark-based navigation systems. Landmark-based navigation systems serve to complement the data obtained from the main perception and environment sensing system to determine the vehicle positioning within a known environment or to determine the most efficient route to a specific landmark; this can be done by establishing spatial relationships, with their corresponding constraints. Constraints can be established through the measurement of the spatial relationship between a known landmark and the vehicle itself or through a geometric relationship such as collinearity or coplanarity [42, 43].

Vision systems are often implemented to recognize landmarks in an urban environment; the algorithms used for this purpose execute feature extractions using resources such as the bag-of-words framework. Data obtained from the vision system can be then compared to images in a cloud-based database server to recognize a specific landmark. The reliability of this method and the integrity of the data reported to the server can be further improved through the integration of other sensor systems; an example would be the validation of the outlines provided by the vision system through correction using LiDAR point clouds [43, 44].

It must be noted that the main limitation of autonomous navigation using a landmark-based system is the lack of a comprehensive and readily available database of every single landmark in a city and the fact that its query accuracy relies solely on the validity and actualization of entries in the database. While companies such as Google and Waze resort to user-fed databases that are continuously being updated, it is important to note that they can be immediately accurate only up to a certain extent because of factors such as human error, deliberated entry of false data due to malicious intent, landmark update time, or simply the sheer amount of landmarks in a city [44].

## 15.4 Localization and Map Building

In an autonomous vehicle, perception is the process to extract different features or elements from the sensors that can describe the environment so we can make a model (map). But what if we have additional information from a knowledge base that can be related to the map of where the robot is located, we can use all this information from the sensors, and the map to determine robot real location. Thus we can describe Mobile Robot Localization as a process used to figure out the pose of a robot; the pose of a robot is the position and orientation, with respect to a model of the environment, and this process is also known as position estimation or position tracking [28].

Mobile Robots have sensors which are usually affected by noise, that is why the robot must be capable of processing uncertainty affected data, and errors detected in sensor readings due to noise are not statistically independent. Actually these errors increase their appearance as long as the number of measurements increases, and the correspondence problem and data association problem raise the question whether the taken measurements are from the same place or object. Correspondence problem can be produced by similarities in the environment or sensor range limits [45].

The concept of robot state is used to describe its pose and also some other magnitudes like robot velocity, sensor biases, and calibration. The map is a description of important features like landmark positions and different physical constraints like obstacles in the environment in which the robot is moving [46].

The process of localization starts by assigning a coordinate system to a map which is called a global frame, another coordinate system that is assigned to a mobile robot whose position and orientation moves along with the robot's pose;

this is called local frame. We have to establish a relationship between the local and the global frames, and this is expressed as a transformation matrix. If we know this coordinate transformation, we can locate position of different objects like landmarks and obstacles with respect to a robot local frame or a global frame, which is a necessary task for robot navigation.

This transformation matrix can be easily obtained if we know the mobile robot’s pose with respect to the global frame as can be shown at Eq. (15.1), as a function of coordinates  $x$ ,  $y$  and orientation  $\theta$ .

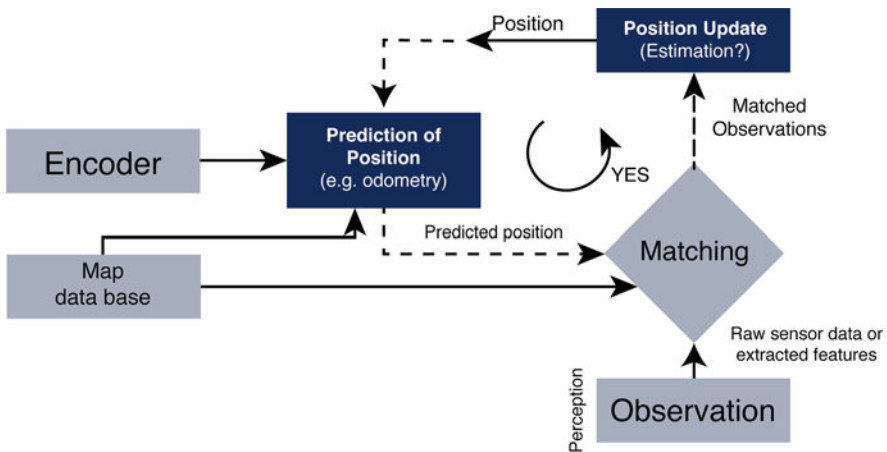
$$M_L^G = (x, y, \theta)^T \tag{15.1}$$

The problem is that robot’s pose cannot be obtained directly from sensors, and it has to be estimated from data measurements taken from the environment [28]. This is because all measurements are affected by sensors noise, aliasing, and also the actuators that carry some noise and inaccuracies.

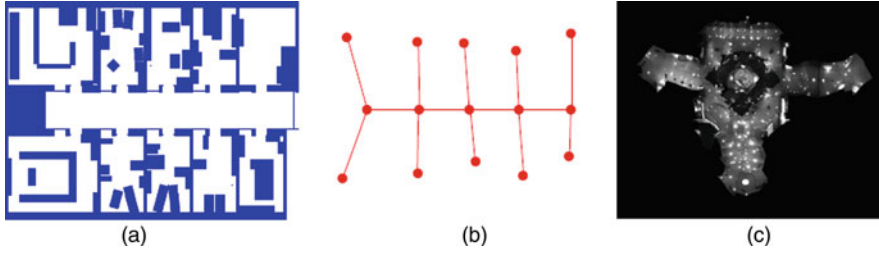
A general schematic for the mobile robot localization is presented in Fig. 15.21. Position estimation is carried out by probabilistic algorithms, and these types of processes have received a great research attention over the last years [47].

Localization is enforced with a collection of map representations; some of these are [28]:

- (a) Feature-based maps. Specify the shape of the environment at the specific locations, this means the location of the objects contained in the maps.
- (b) Location-based maps. These are volumetrics, they offer a label for any location in the environment, also have information about objects and absences of objects contained in the map.



**Fig. 15.21** Schematic for mobile robot localization process [47]



**Fig. 15.22** Examples of maps used for robot localization process: (a) 2D metric map, (b) graph topological map, and (c) mosaic from ceiling [28]

(c) Occupancy grid maps. This is a type of location-based map, they assign a binary value to a x-y coordinate grid associated in order to know if it is occupied with an object. Some instances used by this representation are shown at Fig. 15.22: a 2D metric map at (a), Graph-like topological map at (b), and Image Mosaic from ceiling at (c).

There are many types of maps used for localization and navigation algorithms. The localization problem assumes that an accurate map is available.

Localization problem challenge can be classified by the nature of the environment and the initial knowledge that a robot possesses about its localization. So we present four different criteria considered to solve the location problem [28].

- (a) Local and global localization. Localization problems can be classified by the type of information that is available initially or during navigation.
- Local localization assumes that the initial pose of robot is known so the difference between real robot pose and calculated is small.
  - Global localization. The initial pose of the robot is unknown. The robot is placed somewhere in its environment but does not know where.
- (b) Static and dynamic environments. If the robot is the only object moving on its environment, then it is called static; hence, dynamic environments consist of multiple objects moving during robot navigation.
- (c) Passive and active localization approaches.
- Passive localization approach consists of a robot navigation which is not aimed to improve localization algorithm.
  - Active localization approach controls the robot navigation in order to minimize localization error, but in practice these algorithms combine designated tasks fulfillment with localization goals.
- (d) Single and multi-robot localization.

## 15.4.1 Mapping Sensors

### 15.4.1.1 LiDAR Sensor

As seen in Sect. 15.3.1, a LiDAR (also known as optical radar) is also known as a ToF sensor that consists of a transmitter that illuminates a target with a laser beam, and there is a receiver capable of detecting the returning component of light coaxial with the transmitted light [47]. The transmitter produces a pulse of light aimed at the environment; if there is a target, a component of the pulse returns, a mechanical mirror system is implemented to sweep the light beams and scan a required space which can be planar or three dimensional. The range between source and target can be measured by different methods:

- (a) ToF, this is sending the pulse beam and detecting the time it takes to be reflected and returned to the receiver. The range  $R$  is calculated by the following expression at Eq. (15.2) [48].

$$R = \frac{1}{2}c\Delta t \quad (15.2)$$

where  $c$  is the speed of light and  $\Delta t$  is the ToF of the pulse

- (b) Measure the beat frequency between a Frequency Modulated Continuous Wave (FMCW) and its received reflection [47].  
 (c) Measure the phase shift of the reflected light. Phase shift laser scanners are more accurate, but range is shorter [48].

The phase shift measurement sensor transmits an amplitude modulated of light at a known frequency, the wavelength of the modulating signal poses the next relationship with the speed of light expressed at Eq. (15.3) [47].

$$c = f \cdot \lambda \quad (15.3)$$

where  $c$  = speed of light

$\lambda$  = wavelength of the modulating signal

$f$  = modulating frequency

The total distance covered by the emitted light is given by Eq. (15.4):

$$D' = L + 2D = L + \frac{\theta}{2\pi}\lambda \quad (15.4)$$

where  $D'$  = total distance covered by the emitted light

$L$  = distance between transmitter and phase measurement sensor

$D$  = distance from beam splitter and the target

$\theta$  = electronically measured phase difference

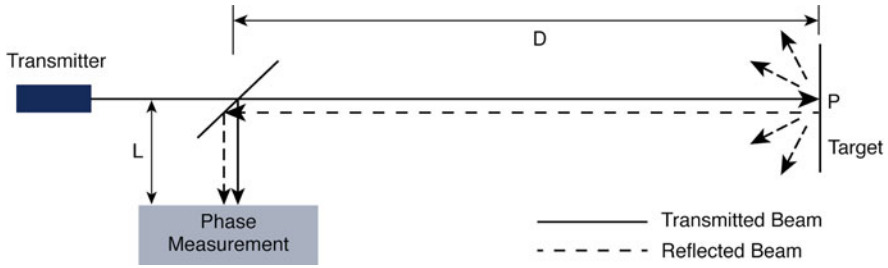


Fig. 15.23 Phase shift measurement [47]

The required distance beam between the beam splitter and the target is given by Eq. (15.5):

$$D = \frac{\lambda}{4\pi} \theta \quad (15.5)$$

Figure 15.23 shows a schematic of a laser range measurement system using the method of phase shift; a light beam is collimated and transmitted from the transmitter, and hits the target point P in the environment and is reflected back to a receiver.

Multi-beam laser scanners are used in robot mobile applications; these systems use an array of sensor applying the method of ToF measurement, and this array of laser beams scan and measure simultaneously and generate in parallel a wide amount of coordinate points which is called point cloud and represent the environment with high fidelity. As a consequence multi-beam laser scanners are used to capture wide regions and landscapes and are used for mapping applications. The laser pulse repetition rate (RRR), in combination with the scanning mirror deflecting pattern, determines the LiDAR data collection rate [48].

#### 15.4.1.2 Kinect Sensor

Laser scanners have obtained great relevance, making 3D models of the environment. Some parameters of laser scanners are very important to differentiate one system or another, parameters such as measurement rate, range, and accuracy are considered for this task. Autonomous mobile vehicles use laser scanners to obtain a model of their environment. Since this task is performed in real time, they need a high scanning speed and also require a medium range (30 m maximum) and low accuracy (3–5 cm) capabilities. Video Game industry has made a contribution to the autonomous robots, since they have developed the so-called gaming sensors, which are devices that have a high measurement speed and an intermediate range (between 1 and 5 m) [49].

Asus and Microsoft companies have developed Xtion and Kinect systems, respectively; both use the triangulation method for measuring targets and have become very popular due to the large number of developers who have done projects with these systems and have extended the potential of their applications to other fields

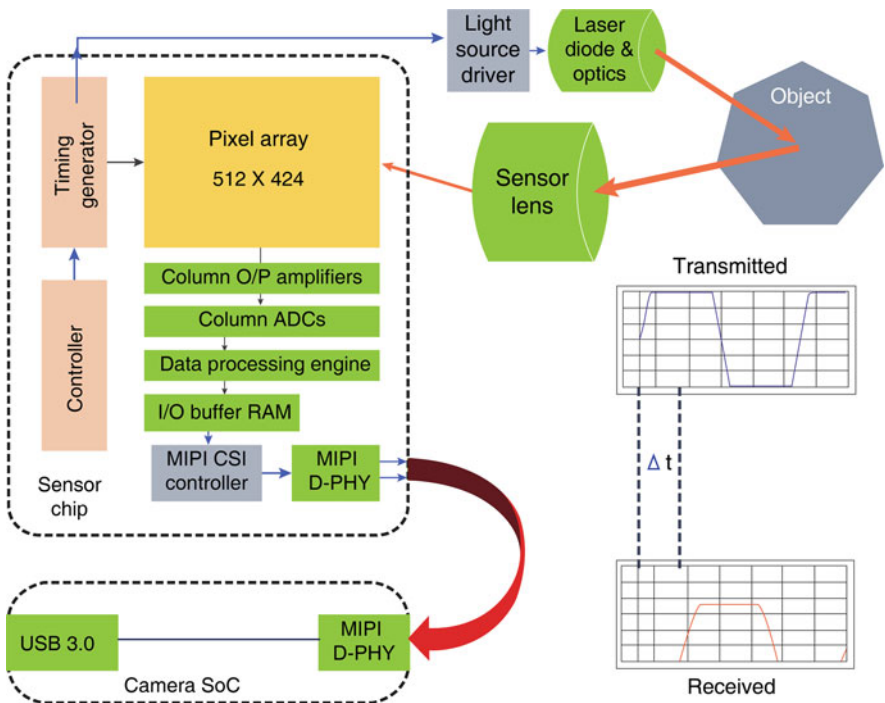
different from entertainment, such as facial recognition, virtual learning, forensic science, and of course navigation of autonomous vehicles [49].

The “range sensors” provide a way to capture measurements of space and objects from mobile robot’s environment. Some important requirements must be considered to select this type of sensors, that is, size, weight, resolution, refresh rate, field of view, and robustness to external light conditions.

Kinect is a high-speed sensor for 3D measurement applications and has a fairly accurate resolution and many applications in robotics and navigation. The first generation of Kinect uses triangulation technologies to make the measurement; however, the fundamental structure of this technology is not suitable in sunlight conditions, and this limits its scope only for indoor application [50].

A second generation of these sensors arose with the appearance of the Kinect v2, which is based on the ToF measurement principles (like LiDAR), which allows it to be used outdoors and in sunlight. It offers higher resolution and a larger field of view compared to that of the first generation.

The Kinect v2 depth sensor has a strobe IR light which illuminates the scene and is reflected by obstacles, and the ToF for each pixel is registered by an IR camera. Figure 15.24 shows the whole system, which includes illumination, sensor



**Fig. 15.24** 3D image perception system [51]. The system comprises the sensor chip, camera SoC, illumination, and sensor optics

optics, sensor chip, and a camera with System on a Chip (SoC) technology. The SoC function is to communicate with the computer or Xbox for gaming applications. The ToF measurement system sends a modulated square wave signal to the light source driver in order to measure the time it takes to travel from diode optics to the target and returns back to the sensor optics, and this is used to calculate distance. The timing generator creates a modulation square signal, which is used to synchronize light source driver transmitter generation and the pixel array receiver detection. Light hits the object and comes back to the sensor optics in time  $\Delta t$ , the system calculates  $\Delta t$  estimating phase shift between received light signal at each pixel and the known synchronization signal (indirect ToF) [51]. Due to the intensity of the signal emitted to determine the ToF, the sensor can be used outdoors.

A mathematical model of noise can be obtained, which can be used in post-processing tasks in the depth image filtering stages for any application. Different measured characteristic parameters can be analyzed to determine the qualities of Kinect v2 in robot navigation, for example, systematic and non-systematic error, measurement capabilities in short ranges, and the influence of ambient light for indoor, cloudy, and direct sunlight applications. The noise model can be a function of the distance of measurement and the angle of the observed figure; however, the model should include conditions of extreme solar luminosity, overcast, and interiors [50].

In order to integrate Kinect v2 to a robotic system, it is necessary to associate a coordinate frame system with the depth measurements in terms of a global coordinate frame which describes the environment or a local coordinate frame associated with robot pose. Measurements collected are processed and can be rendered as robust maps which have a lot of data which describes the environment and objects around.

## **15.4.2 Localization Sensors**

### **15.4.2.1 Wheel Encoders**

Wheel/motor sensor has a wide history being used to measure internal state and dynamics of motor powered systems. These types of sensors offer an excellent resolution at a very low cost. These sensors are usually located at the shaft of the motor drive or at a steering mechanism attached to the motor.

Optical encoders can be classified due to the way it presents the data in:

- (a) Absolute encoders. Represent a position with a unique digital code, this means that if the encoder has a resolution up to 16 bits, has a span of  $2^{16}$  different codes to represent a physical position.
- (b) Incremental encoder. Generates an output signal every time the mechanism moves, the number of signals per revolution or in a span of distance defines the resolution of the device.



In mobile robotics, incremental encoders are the most common device, used to measure speed and position within a motor drive or at the shaft of a wheel, steering mechanism, or another joint mechanism. The sensors are proprioceptive so their estimation of position has the best performance in the reference frame of the robot and need corrections when applied to mobile robot localization.

An optical encoder produces a number of pulses sine or square for each shaft revolution, and it has the next elements.

- Illumination source
- A fixed grating that masks the light
- A rotary disc with a fine optical grid that rotates with the shaft
- A fixed optical detector

As the motor shaft moves, the amount of light varies based on the alignment of fixed and moving gratings. In robotics, the resulting sine wave is transformed into a square wave.

As the motor shaft moves, the amount of light that reaches the optical detectors varies based on the alignment of the fixed and moving grids. In the robotics, the resulting sine wave is transformed into a square wave using a threshold value to differentiate the values that represent the light and dark states [47].

Resolution is measured in pulses per revolution (PPR) or cycles per revolution (CPR). A typical encoder in mobile robotic applications may have 2000 PPR, while the optical encoder industry can reach up to 10,000 PPR sensors. Also, the sample rate needed for detecting shaft spin's speed can easily be achieved by commercial encoders, so there is no bandwidth limitation for mobile robot applications.

Quadrature encoders are another version of these sensors modified in order to obtain more data from an incremental encoder. This device has two pairs of emitter-detector instead of one, both detectors are  $90^\circ$  shifted with respect to each other in terms of the rotary disc grid. The resulting output signal is a pair of square wave shifted the same  $90^\circ$  as can be seen in Fig. 15.25, both signals receive the name channel A and channel B. A third pair emitter-detector is included in order to detect one unique position called index, which is used as a reference to count revolutions or finding home position. As it can be seen in Fig. 15.25, this index sensor pair has only one slot clear position in the outer track, and thus, it is only one pulse in the whole

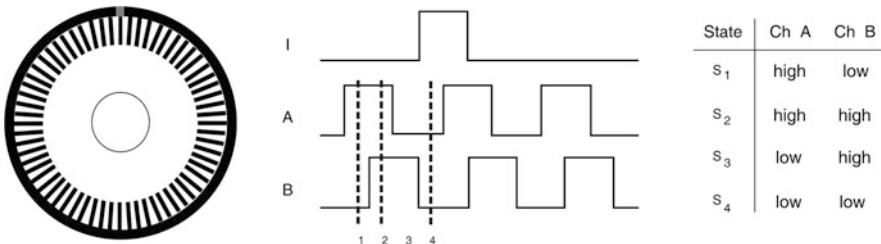


Fig. 15.25 Quadrature optical incremental encoder [47]

turn. Channel A and channel B signals provide more information such as direction of rotation (if we detect which signal rises first). In addition, a different combination of channel states causes different detectability states which increase resolution by four. This means that a sensor with 2000 PPR hence in quadrature yields 8000 PPR.

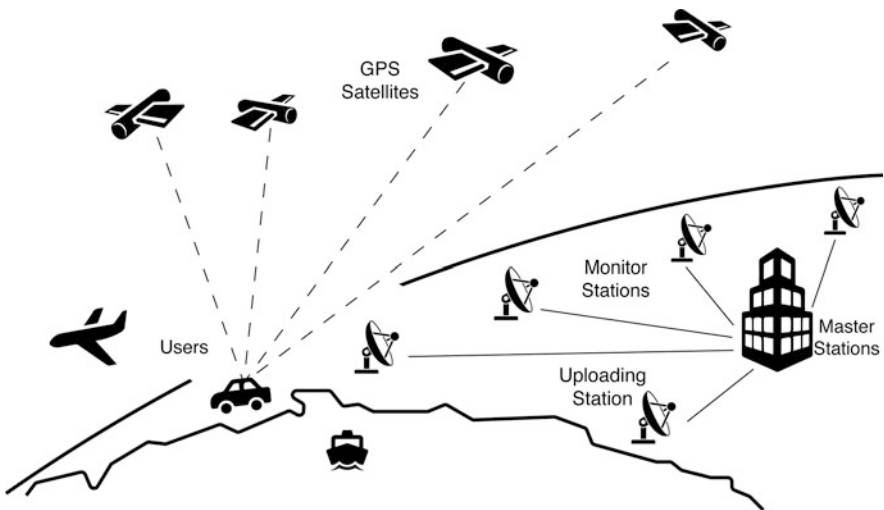
The systematic errors and the cross sensitivity inherent to encoders can be eliminated using engineering tools such as filters and probabilistic methods. The accuracy of optical encoders is usually considered 100% for mobile robot applications, and any error due to any optical encoder failure is minimized by errors caused by motor shaft or actuator.

#### 15.4.2.2 Global Positioning System (GPS)

Historically humans have used beacons, like stars, mountains, guide posts, and lighthouses for navigation, a modern approach involves the use of emitters located far away from a receiver which is situated at the mobile vehicle. This sensory setup makes available accurate outdoor localization, with some limitations due to the nature of the technology used.

GPS is a modern beacon system instrument used nowadays for outdoor navigation and localization of air and land vehicles.

Twenty-four satellites are available for free GPS service, and they orbit every 12 h. Four satellites are located in each of six inclined planes,  $55^\circ$  with respect to the plane of the equator [47] (see Fig. 15.26).



**Fig. 15.26** Calculation of position and direction based on GPS usually requires more than three satellites [47]

GPS are passive and exteroceptive sensors. Satellites with GPS service are synchronized, so their transmission signals are sent at the same time. A GPS receiver reads the transmission of two or more satellites, the time difference between these transmissions can be used as an indicator to the receiver about its relative distance to each satellite. By combining the information of the position of each satellite and the delay of arrival times, the receiver can calculate its own position.

This operation requires only three satellites, but as long as there are more transmitting to the receiver, position calculus becomes more accurate. The arrival time delay ranges are of the order of nanoseconds; that is why signals from satellites must be perfectly synchronized. In order to achieve this, satellite timing register is refreshed by ground stations routinely and also an atomic clock is implemented on satellite's circuitry [47].

GPS receiver clock timing is Quartz crystal based, that is why often needs more than three satellites to have an acceptable location reading, that is the main reason for GPS limitations on cramped spaces like those surrounded by tall buildings, big trees, or mountains and places where reading more than three signals at the same time becomes impossible, and position calculus is inaccurate, that is also why indoor applications for GPS fail to provide a reliable position sensing.

GPS performance is also affected by other different factors like the orbital path which is not a straight line, but a curve, and therefore, resolution is not uniform on all earth regions, having a lot of variations and uncertainties away from the terrestrial equator.

GPS satellite information can be obtained by implementing different techniques in order to get better resolution. Some of these techniques are pseudo-range which usually has a resolution of 15 m; another technique is the so-called differential GPS (DGPS), which uses a second receiver located at a reference place to correct position and can have resolution of less than 1 m; and one last technique is used measuring phase of the carrier signal from satellites transmission, this GPS receiver can achieve 1 cm resolution for point position [47].

One last consideration must be taken into account when using GNSS receiver with automated mobile robots, which is update rates. Usually 5Hz GNSS updates rates are obtained, that is why GNSS are used with other types of sensor and navigation algorithms.

### ***15.4.3 Navigation Control***

#### **15.4.3.1 Simultaneous Localization and Mapping (SLAM)**

On a day-to-day basis, we make use of navigation control when we drive our cars or when we navigate inside of schools or buildings. But unless we have a map, we must always do an exploration previously. Nowadays we have applications to guide us anywhere in the world (such as Google Maps and Waze). Accordingly, autonomous

vehicles need these requirements too; they need to have a predetermined map of their work area or they need to create one while exploring their environments.

The process of simultaneous localization and mapping (SLAM) is sometimes called the concurrent mapping and localization (CML) process. This is the continuous construction of a map and the computation of the state of the robot within its environment [46].

The maps give a human operator an intuitive perception of the environment, but also it is essential to create location maps for the navigation control of autonomous robots, several types of maps can be generated (e.g., feature based or location based) [28]. These can be used for route program and the correction of the calculation error produced by estimating the state of the robot. When the hardware and software are more complex, robots can even be programmed to make their own decisions in real time in order to avoid any collision or path planning. If we do not have a map, the navigation estimation would lose its path and the vehicle would drift. But if we have a map, the robot can correct its location error by returning to the same areas; this process is also called loop closure [46].

SLAM can be used in applications in which a map is not available and requires to be created, that is, in SLAM the trajectory of the vehicle and the location of all landmarks are estimated online without the need for any a priori knowledge of location [52]. Sometimes SLAM may not be required, like in applications in which an accurate prebuilt map describing fixed physical constraints of the environment is provided, that is, an autonomous vehicle operating in an indoor facility with a manually built map and also in applications in which the robot has access to a GPS (the GPS satellites can be considered as moving beacons at known locations) [46].

In most of the projects, different types of sensors were used to cover the widest range of visibility and scope possible, from ultrasonic sensors to RGB-D cameras, LiDAR, and IMU such as accelerometers, encoders, and GPS. All of these are necessary to be able to map and make decisions for the robot in navigation control, in addition to having algorithms to represent the 3D and 6D maps for vehicles.

The autonomous navigation of a robot is a complicated task which leads to several problems that need to be solved. Nowadays, the articles written on projects or advances on this subject are more concurrent specific, and with more diversity on how to apply the hardware to the software in order to achieve a great improvement in performance. Hence, we will see some case projects related to an area as important as is the control of navigation for autonomous robots.

In most of the investigations about autonomous robots' navigation, the SLAM used is made by LiDAR (Fig. 15.27). Its use can be divided into two categories: first, to make an initial map sample of the area where the robots will be used and, second, also [53] can be used on top of a robot to continue building maps while the robot interacts with its environment and, thus, be able to have a better response to certain situations. Other projects generate different types of mappings with different resolutions for certain specific tasks that they want to solve. For example, in [53], to maintain high performance and low memory consumption, they used multi-resolution maps, such as maps based on octrees (Octomap) which are a structure of information for multiple storage resolutions in 3D, how you can see at Fig. 15.28.

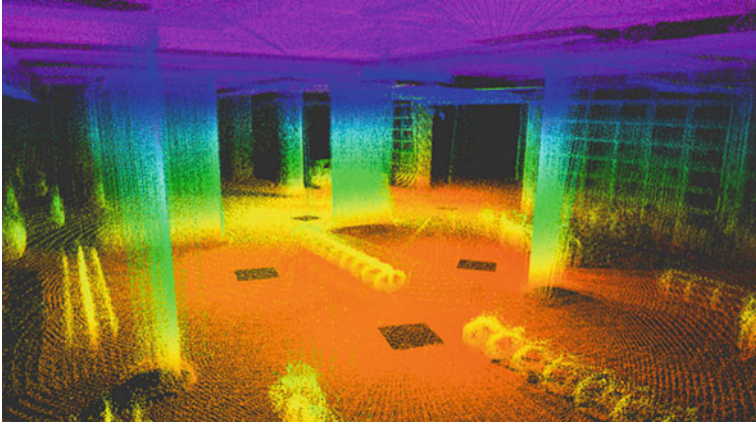


Fig. 15.27 Aligned scans from parking garage [53]

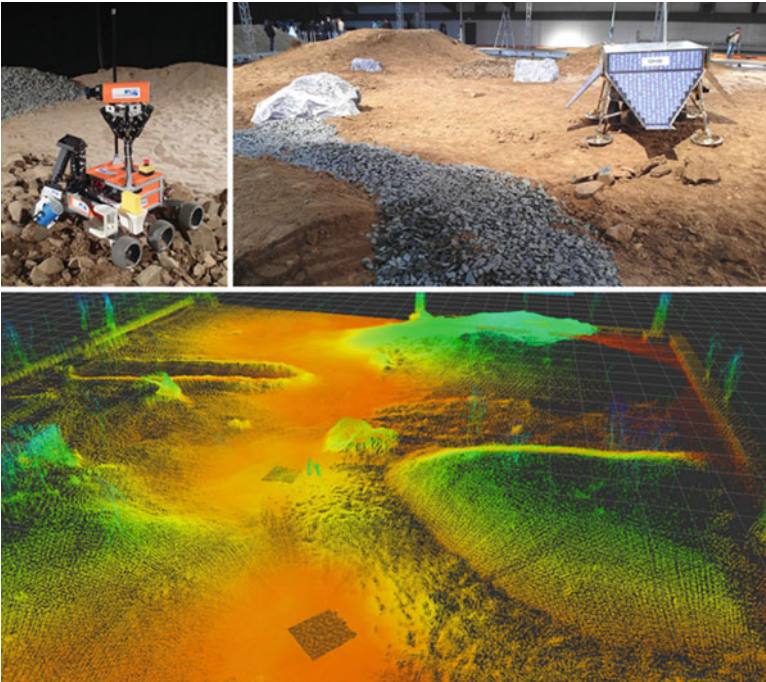
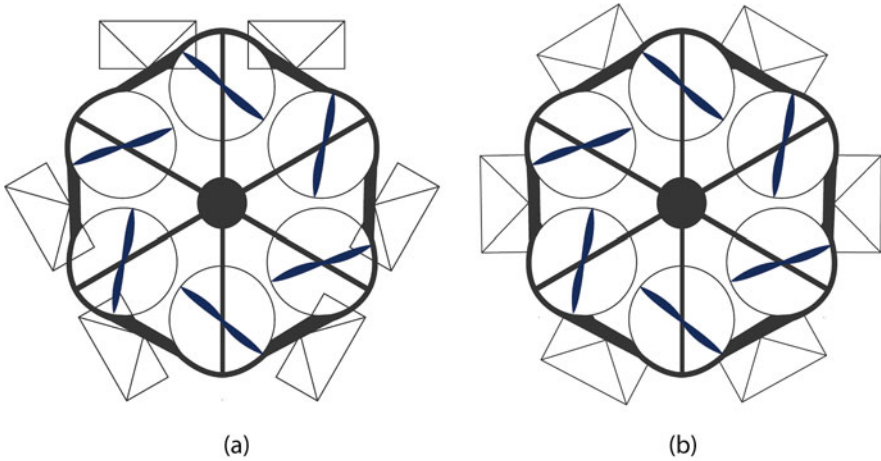
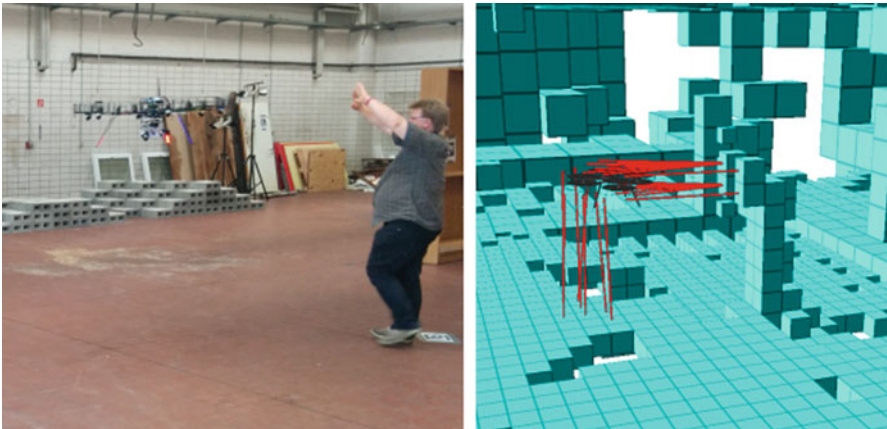


Fig. 15.28 (a) Explorer, Schadler team robot for DLR SpaceBot Cup (upper left image). (b) SpaceBot Cup Arena (upper right image). (c) 3D scans of the arena taken by Schadler (lower image) [53]

In other autonomous robots, we can find [54] cameras and IMUs (barometers, gyroscopes, accelerometer, and Global Navigation Satellite System (GNSS)) in addition to having a LiDAR. The cameras can rotate to have an omnidirectional vision, or stereo configurations as can be seen in Fig. 15.29, depending on their locations in the vehicle. All of this to be able to obtain 6D maps based on maps in SLAMs. In environments where it is not possible to use GNSS, all of the other components can give estimated location coordinates to be able to general mapping in 6D (Fig. 15.30).



**Fig. 15.29** Mounting of the cameras in micro aerial vehicles (MAV). (a) Triple stereo configuration. (b) Omnidirectional configuration [54]



**Fig. 15.30** (a) Picture taken when testing the MAV with a person. (b) Scan from the MAV, where the MAV is pushed away from approaching a person [54]

Once having the mapping of the area where the tests and activities are to be carried out, in addition to having SLAM as well, a big problem that presents itself in an autonomous robot will arise since the pre-established map will be compared with the route in real time. Therefore, obstacles or uncertain situations can be presented, and it is here when it is of the utmost importance for the robot to have a navigation control, so that it will be able to dodge the aforementioned obstacles and to select other routes with better times or that are clearer. For decision-making, various algorithms and logics accompanied by hardware in order to be able to enter data for feedback are incorporated. And so, all this can be used in various ways, such as in 2D maps, 3D, and 6D being the main modes of mapping. This means that the equations that are designed to be used in robots are different. The design of their flow diagrams also has an effect consequently; these two are going to determine all of their model decisions. Currently, there is a project [55] in which the images captured by the cameras of the robots pass through filters which can see each step of the robot. With the application of equations, the free areas that lie ahead can be predicted. In addition, a 2D map is generated in which their layers are completely connected. For this reason, their decision-making is based on networks of value interactions that work, in turn, to a neural network (Fig. 15.31).

In another case [55], they generate SLAM to see their environment in real time, thereby knowing if there are variations, such as temporary or permanent obstacles, changes in their work environment, with their predetermined mapping or with previous interactions with their journey. In addition, they make cells (grid cell) in 2D and group them to evaluate the drivable areas, all of this to avoid complicated areas where it is not optimal for the robot to move (Fig. 15.32). In this last case [54],

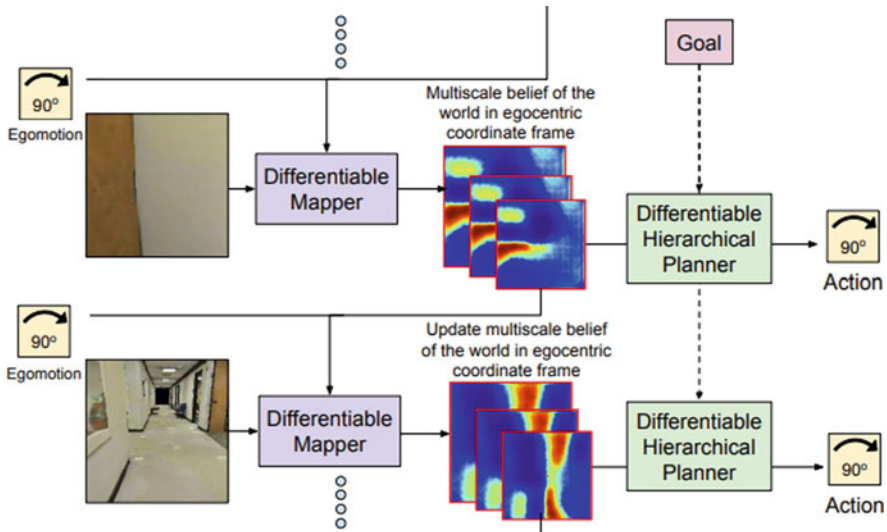


Fig. 15.31 Overall network architecture from S. Gupta [55]

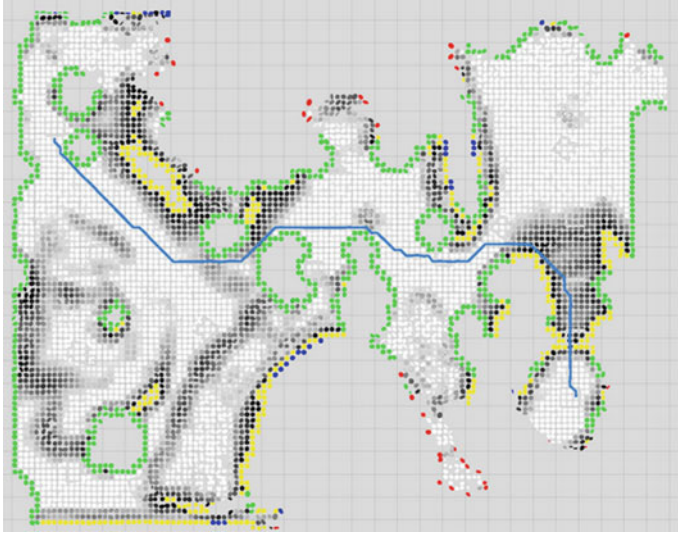


Fig. 15.32 Navigation planning [53]

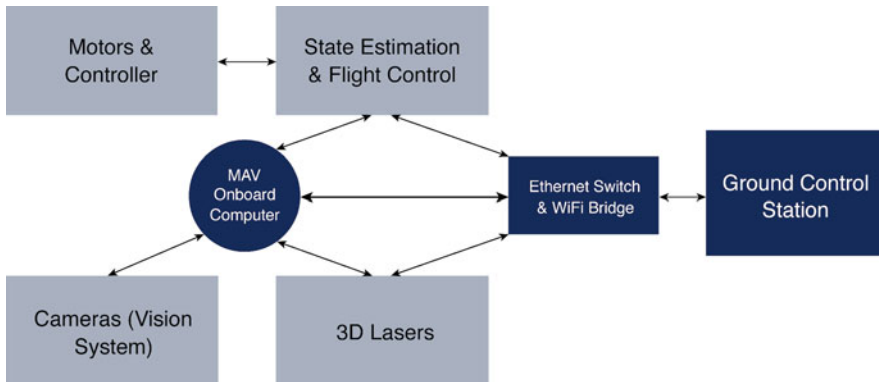


Fig. 15.33 Overall schematic for MAV [54]

an autonomous MAV uses cameras to classify objects but also uses a SLAM (3D lasers) to create 3D maps. Also, with both technologies, they can create environments in 6D, which allows them to identify the obstacles to redo a new route to reach the destination. Apart from this, GNSS is used for outdoor environments, but for interior use, a visual odometry (VO) is used as well as components that integrate an IMU in order to be able to know their location in the 3D and 6D planes (state estimation and flight control). All of this information passes through usb hub 3.0 and 2.0 for it to be processed in a computer center inside the vehicle and to send information to another computer via WiFi for monitoring (Fig. 15.33), which shows an overall schematic.



## 15.5 Path Planning

When we address any topic related to autonomous transportation, often many doubts arise, that is because there are several factors that must be taken into consideration in order to successfully reach our desired destination; being the motion or path planning, one of those factors, and that is what this section is oriented to.

As humans, most of the time, we do not even need to worry about getting from point A to point B, as a result of our capabilities in identifying the answers to the questions: where am I? Where do I want to go? How do I get there? And what are the obstacles or dangers in the path to my objective?

In order for a robot to answer the questions shown previously, a certain kind of intelligence might be integrated into it; in consequence, many kinds of research have been done in subjects such as mapping, localization, and path planning. The objective of this last subject is the minimization of the distance the vehicle needs to travel, and to accomplish that, an efficient path planning algorithm is necessary.

Some years ago the computational power available could not handle the requirements for online path planning algorithms, and researches opted for rule-based algorithms that gave good results in indoor controlled environments [56].

### 15.5.1 Algorithms

#### 15.5.1.1 A\* Algorithm

The A\* algorithm belongs to the graph search group, and it shares several similarities with the Dijkstra's algorithm. A\* is one of the most efficient algorithms in solving the problem of path planning; however, in scenarios with a high density of obstacles, it might not be the ideal algorithm to implement due to its high time consumption when it is dealing with a hard to process situation.

$$f(n) = g(n) + h(n) \quad (15.6)$$

A\* is based on the evaluation function presented in Fig. 15.34, where  $g(n)$  is the distance from the current node to the node  $n$ . On the other hand,  $h(n)$  is the distance from our goal position to the node  $n$ . The distance can often be called "cost" or "heuristic."

The way A\* works is simple yet brilliant. The algorithm starts by identifying the nodes that already have a direct path to the starting position, following the example in Fig. 15.34. Those nodes would be  $F$ ,  $E$ ,  $B$ , and  $D$ , then the algorithm evaluates the already mentioned nodes with Eq. (15.6) (see Table 15.1).

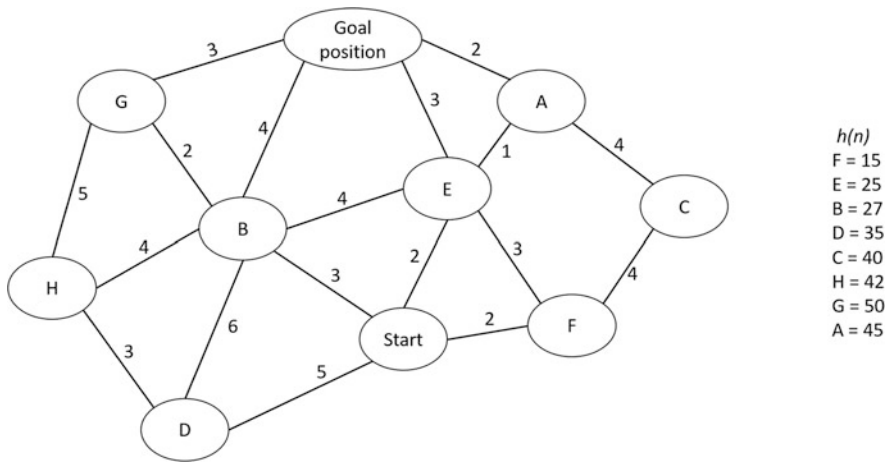


Fig. 15.34 A\* algorithm example

Table 15.1 A\* algorithm node evaluation example

$f(F) = 2 + 15 = 17$
$f(E) = 2 + 25 = 27$
$f(B) = 3 + 27 = 30$
$f(D) = 5 + 35 = 40$

### 15.5.1.2 Field D\* Algorithm

In mobile robot navigation, a representation of the environment is needed, generally divided in cells. Depending on the environment, the representation could be binary, obstacle, or free, or it could have an associated cost for each cell,  $g(s)$ . Most algorithms like Dijkstra, A\*, or D\* are limited by the small set of possible transitions; they plan from center of cell to center of cell resulting in paths that are not optimal and difficult to follow in practice [57].

Traditional algorithms compute the path cost assuming that the only possible transition from one cell to another is a straight line from the cell to one of its neighbors. If this restriction is relaxed and the path can go from the cell to any point in the border of the neighbor cell, a minimal cost trajectory can be calculated. Unfortunately, there are an infinite number of points, and calculation of all the possible trajectories is not possible.

It is possible to provide an approximation for each boundary point by using interpolation. In order to do this, it is necessary to assign nodes to the corner of each cell, with this the cost of traversing two equal length segments of an edge will be the same, this is shown in Fig. 15.35. This solves the cost problem when a segment crosses two cells, each with different cost. The nodes in the graph are used as sample points of a continuous cost field. The optimal path from a node must pass through an edge connecting two consecutive neighbors.

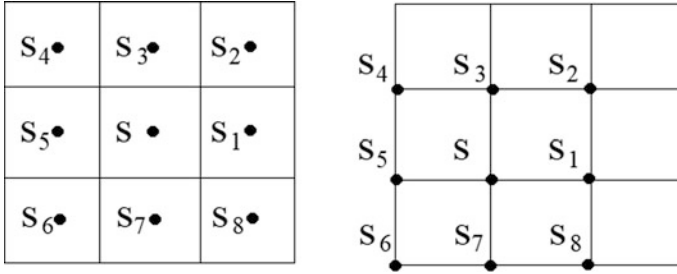


Fig. 15.35 Solution of the cost problem

To calculate the cost efficiently, assume the path cost of any point  $s_y$  residing on the edge between nodes  $s_1$  and  $s_2$ , which is a linear combination of their cost.

$$g(s_y) = yg(s_1) + (1 - y)g(s_2) \tag{15.7}$$

Equation (15.7) is not perfect, and the path cost may not be a linear combination of  $g(s_1)$  and  $g(s_2)$ , but it is sufficiently exact to be practical.

The path cost, given  $s_1$  and  $s_2$  and cell costs  $b$  and  $c$ , can be calculated by Eq. (15.8):

$$\min \left[ bx + c \sqrt{(1 - x)^2 + y^2} + yg(s_2) + (1 - y)g(s_1) \right] \tag{15.8}$$

where  $x$  is the distance traveled along the bottom of cell  $s$  before turning the center cell to reach the right side at a distance  $y$ .

Because a linear interpolation is used, it is less expensive to cut through the cell than following the boundaries. If there is a component of the cheapest solution that goes through the center of the cell, then it will be as large as possible forcing  $x = 0$  and  $y = 1$ . If there is no component through the center, then  $y = 0$ .

Assuming that the optimal path is as shown in Fig. 15.36a, it travels along  $x$  axis to a certain point and then crosses diagonally to a point  $y$  and finally to the upper right node. Clearly this path is shorter than going horizontal all the way to lower right node and then up to upper right node. Scaling the resulting triangle so that the upper vertex is now located in the upper right node, maintaining the slope will be shorter than the previous path as seen in Fig. 15.36b.

Thus, the path will travel a distance over the  $x$  axes and then go straight to the upper right node, or it can go directly from the origin node diagonally to some point on the right edge and then up to the upper right node depending on the costs  $b$  and  $c$ .

Once the cost of the path from the initial position to the goal is calculated, the path is determined by interactively computing the cell boundary point to the next. Because of the interpolation technique, it is possible to calculate the cost of any point inside a cell, not just the corners.

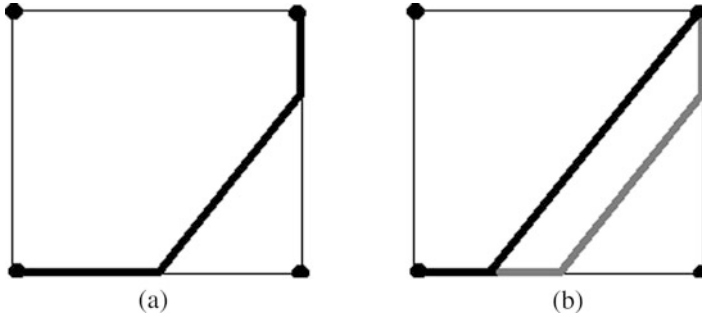


Fig. 15.36 (a) Optimal shortest path solution. (b) Simplified path with same slope solution

## 15.6 Case Study: Intelligent Transportation Scheme for Autonomous Vehicles in Smart Campus

Nowadays, diverse autonomous vehicle designs are used to assist in various daily tasks, for example, in smart farming, smart industry, among others. The self-driving vehicles positively impact on economy and environment, as well as the completion of tasks efficiently. Similarly, the use of autonomous vehicles on Smart Campus could increase student and worker satisfaction at universities as it is the case of the intelligent transportations scheme for autonomous vehicles in CETYS University [58].

To introduce this case study, it is essential to understand the context. CETYS University Mexicali Campus is located at the capital city of the Mexican State of Baja California, which is a US–Mexico border city. Mexicali city has a warm and arid climate; in consequence, walking in summer could be very dangerous. According to the National Institute of Ecology, the fatality rates caused by heat waves in Mexico were presented in Mexicali [58]. Likewise, the Mexicali’s General Hospital reported 72 patients with heat waves symptoms, and 23 of them died during the period from 2006 to 2010 [59].

Consequently, autonomous vehicles in the campus will possibly be a solution to reduce the risk that heat waves affect the students, professors, and administrative personnel during their movement inside the university. Additionally, above all population inside the campus, a significant number are elderly people, and studies have shown that elderly population is more vulnerable to heat waves and elevated temperatures [60, 64], given the previously mentioned statistics.

Moreover, seeking the well-being of people with mobility or visual impairment in the campus, self-driving vehicles benefit their transportation inside the university. As the parking and buildings inside the campus are separated by long distances, the possibility to have a qualified transportation system improves the life quality and inclusivity inside the campus.

The previous context description stands for the motivation to develop an intelligent transportation system implemented in an automobile structure capable to transport passengers between buildings inside the campus and from the parking

to a building. The design of the vehicle includes the artfulness to avoid obstacles (i.e., people walking through the path). Hence, the design of the vehicles included the development of a 3D vision system [61] capable of detecting the road, potential static, and dynamic obstacles (or dangerous conditions), among other variables. Then, design an autonomous navigation system capable of safely traversing a path from point A to B using the 3D vision system data and re-route the vehicle if needed to achieve the desired location. Also, build a sustainable electrical system for the vehicle, with a battery capable of recharging itself through solar energy. Finally, design a network infrastructure to suffice the whole communication scheme and fulfill the requirements of a Smart Campus.

The term Smart Campus refers to a university campus capable to keep all the data infrastructure and offer different services to the students and to all people inside the campus. Hence, to develop a smart agent inside a smart network, some technologies should be adopted as a standard, for example, the sample rate for sensors, the data communication scheme, how systems interact with each other, how all the microcontrollers behave as master and slave depending on the network itself, etc. One of the agents interacting in this Smart Campus would be the autonomous vehicle. To achieve an effective interaction inside the system, the vehicle must be capable to respond to user's requests, to navigate avoiding obstacles and leave the passengers at their selected endpoint. The steps taken for the design of the autonomous vehicle are described in overall terms.

### ***15.6.1 Applied Simultaneous Localization and Mapping (SLAM)***

In order to map the surroundings accurately and to properly move along the different route traced by the system, an efficient mapping algorithm known as SLAM (simultaneous localization and mapping) must be implemented (see Fig. 15.37). This algorithm is implemented to provide autonomy to the vehicle.

As Fig. 15.37 shows, the design includes some sensor inputs such as a LIDAR and Kinect Frame, both described in previous sections. Odometry readings change according to the vehicle movement. The uncertainty from these lectures is treated by the odometry update algorithm. The re-observation process is essential to map construction. All the elements improve the accuracy of the vehicle movement.

Figure 15.38 shows the control flow process for the autonomous vehicle. The vehicle starts the movement after getting current location and the calculus of the initial trajectory. The sensors are implemented to detect obstacles. If an obstacle is detected, the vehicle reduces its speed or stops. The map construction process intended for scanning the environment and analyzing data is performed during all the trajectory in order to control correct navigation through map corrections based on the obstacles' position. A GPS is available to give information about vehicle's location and to correct initial trajectory or re-route if necessary. The sensors described

Fig. 15.37 Diagram of a SLAM algorithm [58]

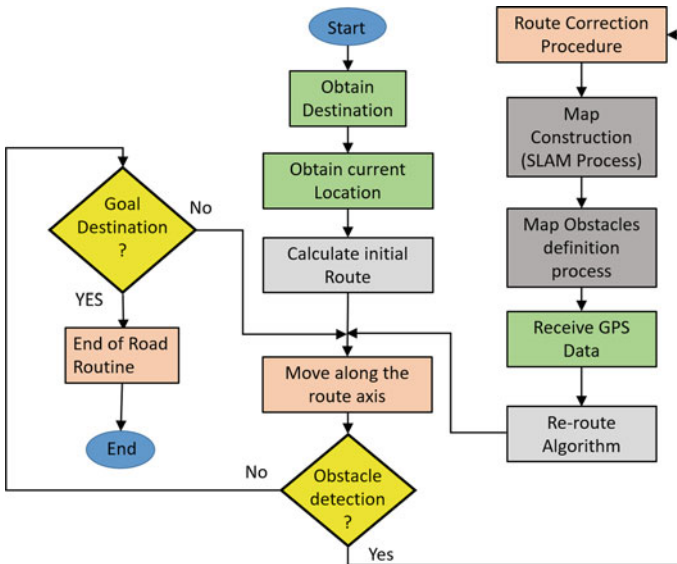
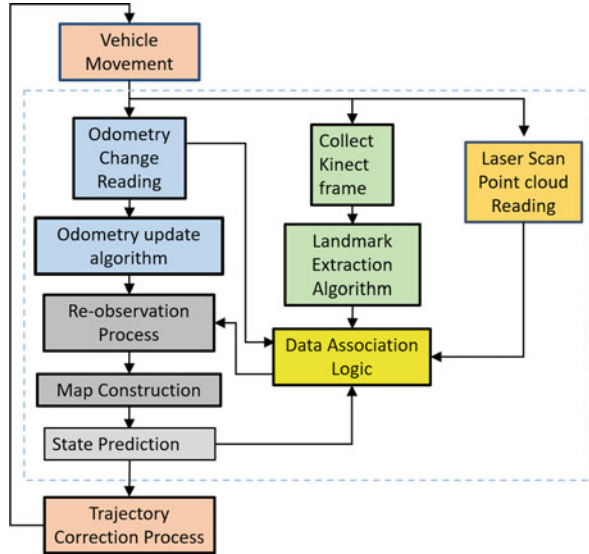


Fig. 15.38 Control flow process for the autonomous vehicle [58]

below should behave synchronously in order to maintain consistency in the whole scheme and for the system to choose the best route to take, and how to re-route correctly taking into account all the obstacles in the vehicle's line of vision.

### 15.6.2 Mechanical Design and Kinematic Model

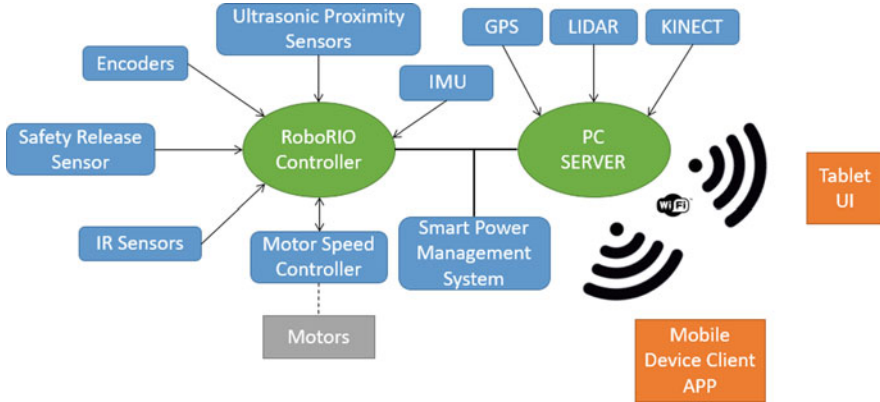
Our mechanical design takes into account the vehicle's chassis, gearboxes, motors, wheels, sensors, batteries, computer, Ethernet switch, cable management, and airflow. The whole autonomous system (vision and control) is mounted on an aluminum chassis with 4// Omni directional wheels in order to satisfy the requirements of forces applied to the base of the vehicle, so all forces can be dissipated and damped. This type of drive allows linear and rotary motion through space which provides the specific advantage of maneuvering in all directions without steering the wheels, which allow for more accurate control and precise movements (see Fig. 15.39).

The proposed architecture scheme is shown in Fig. 15.40. The design achieves the automation goals. The communication between all the elements is established, closing the loop for a complete automation and control scheme.

This case study is an example of how technology contributes to improving the well-being of students, professors, and administrative personnel in universities. Autonomous vehicle must respond needs in Mexicali where the weather conditions affect the health of several people.



**Fig. 15.39** Autonomous vehicle scale prototype [58]



**Fig. 15.40** General interconnection scheme for the autonomous vehicle system [58]

## 15.7 How Innovation in Business Models Will Change the Future of Cars

Electric cars are the future. We all perceive that. It is a topic that resonates in our daily life. But electric cars do not have the same features as gasoline cars, which are more powerful, bigger, heavier and more expensive. Generations in the last one hundred years have seen the evolution of cars, from de T model of Ford to the latest Tesla model. But from the silence generation passing through the baby boomers, generation X, millennials, and finally generation Z, things have changed mainly in the business models, those businesses that are ruled by the consumer profiles of the biggest market segments. Nowadays innovation is changing the rules of the business models. People have the chance to choose between different services online, Amazon, Google, Airbnb, and HostaPet. New business is based on IT platforms where innovation allows the idea to do services online between people around the world. Our generations are in the era of open innovation [3, 62]. This is the time when Amazon has no factories, Airbnb has no hotels, and Uber has no cars. But people can get better services because Internet and smartphones make easier trading.

In the era of open innovation [3, 62], the future of the automobile lays over the profile trends of generation Z. While businesses like Kodak took one hundred years to experience a change in their business model, and lost the battle [62] of the digital camera in the early 2000s, other enterprises like Procter & Gamble have understood market trends to change their business models in order to sustain competitiveness. In that sense, Henry Chesbrough refers to the use of the automobile as an opportunity.



### 15.7.1 *The Misuse of Expensive Vehicles*

Think about the number of hours your vehicle takes to take you anywhere at any time you want during a regular day. Let us say, it is about 2.5 h of use each day. That means you are using your car a little more than 10% of time, but you paid for 100% of the time you really need that good. Assume an average cost of a car as around \$15,000. This means that you paid ten times more of the use of your vehicle, because you only need it 10% of the time. From a business point of view, there is a chance to make business to take advantage of that extra cost between 10% of usage and \$15,000.00.

### 15.7.2 *Generation Z Consumer Profile and the Future of Vehicle*

The business model canvas of Alexander Osterwalder and Yves Pigneur [4] spins around the value proposition for a market segment. Watching consumer profile for generation Z, it can be seen that generation Z or centennials are also considered a better version of millennials. They are digital natives. They do not know a world without Internet. Generation Z will be around 30% of the world population with a very important percentage of purchasing power for 2020 (see Table 15.2). Generations are really fuzzy to define, but this generation locates between 1995 and 2010. For generation Z, value is the number one need as a consumer. This means, as they have many options to choose from, they are not people of one brand. So, for them value is the most important thing about a product or a service. Centennials trust youtubers more than a celebrity, because youtubers are people like them without makeup. It means that they trust authenticity more than appearance. They use to watch at reviews on social media and in general generation Z communicate with their friends, families, and also with brands.

Analyzing the three core concepts of the consumer profile defined by Alexander Osterwalder, it can be expected from the consumer profile of centennials generation that they expect products or services to fulfill its purpose, they appreciate value. So

**Table 15.2** Population resident in the USA in 2017 [63]

Generation	Population (millions)	Percentage	Time range
Greatest	2.57	1	Before 1928
Silent	25.68	8	1928–1945
Baby boomers	73.47	23	1946–1964
X	65.71	20	1965–1980
Millennials	71.86	22	1981–1996
Centennials	86.43	27	1997–2016
	325.72	100	

as they are not loyal to a brand and appreciate authenticity, their consumer profile is practical. So when they spend money, they commit with those products and services that are good about achieving functional jobs over the ones that achieve emotional or social needs. That means, although also as millennials, property is not their first concern, function is the most important thing for centennials as well as for millennials when they buy a car. So in this stage, two of the relevant concepts from the definition of value proposition canvas from Alexander Ostewalder for the consumer profile pointed as pain relievers and gain creators for the generation Z are each day more relevant. From all these generation behavior and trends from a business model point of view, it can be expected that eventually as well as the use of electric cars, new and innovative business models around the consumer profile drivers will be met in the next years.

### 15.7.3 Business Model Canvas for Car to Go

In 2008 Daimler Chrysler launched their program Car2Go ([www.car2go.com](http://www.car2go.com)). Their value proposition (see Table 15.3) for the market segment is as follows: (a) Use by the minute. With car2go, driver is in control of every mile and every minute. Unlike other car sharing services, drivers pay only for how long they actually using the car and (b) one-way trips. Start at point A and end at point B without returning the car to point A. Pick up from the street and get going, (c) on demand. Drivers can get going without prior reservations. Drivers can use their smartphones to find and drive on the spot or reserve for up to 30 min (for free) before beginning.

**Table 15.3** Value proposition canvas [4]

Value proposition	Customer segment
Gain creators Easy localization of an electric vehicle Quick instructions to use vehicle Vehicles registered on Google maps	Gains Quick finding of vehicle Want to pay only by minute Easy finding of the vehicle Smartphone localization
Pain relievers Use of Google map to find easy and quick all the vehicles of the company Battery charged Electric charge stations	Pains Cannot find a car close to the area Easy use of the vehicle Low-battery electric car Sanitized vehicle
Products and services Electric car easy to use rated by minute and registered on Google maps	Customer Jobs Use of a car only for the time the customer needs

### ***15.7.4 Autonomous Car as an Innovative Business Model***

Market trends in recent years formed by the consumer profiles of millennials and centennials open the possibilities for the use of autonomous cars. Evolution of business models based on open services innovation in the era of 5G technology may in the near future set the bases for the use of autonomous electric cars. Clearly, technology has reached a new level with the advent of 5G technology. And innovation on business models, mainly in open services are becoming a very important part of the economies. On this stage, it could happen that new generations will use to buy services online through their 5G technology smartphones, including autonomous electric cars.

Analyzing the value proposition of the business model Car2Go, it could be a natural evolution to scale for autonomous cars. In a first approach, the business model of an autonomous electric car meets all tasks for the profile of the customer. And more, business model of the autonomous car exceeds the expectations defined as gain creators for a value proposition of the Car2Go model. An autonomous car eliminates the need for an easy localization of the vehicle. The autonomous car can go to meet the customer. Being autonomous also discards the need for understanding how to use the vehicle. And of course, autonomous vehicle needs only a button to know where to go. Of course, 5G technology in a smartphone makes easier for customer to command tasks to vehicle. It can be found that autonomous cars can also program electric charge timing.

Finally, it can be concluded that latest technologies and evolution of business models on open services meet profiles of customer segments for new generations.

## **15.8 Conclusions**

In this work, we have reviewed the fundamentals of autonomous vehicles, from automation concepts going through the standard for on-road vehicles defined by the Society of Automotive Engineers, to novel sensors and image processing techniques, including some other commercial products, that is, mobile robots for industry and other small devices as a mobile vacuum. Modernity offers plenty of technology, and beyond an autonomous electric car itself, the disruption line in the advent of autonomous electric cars to be used on a regular basis depends more from the future drivers or even users of autonomous electric cars as a service.

New generations have left the typical profile of a consumer. So, the next step in the business of individual transport is more about the needs of the future driver or user. It seems that at least for high-density population cities, there are still new innovative ways to get from one location to another. Post-centennial generations will use faster technologies and will define market trends. The use of autonomous electric cars in the near future will be a reality when disruptive technologies meet disruptive models for individual transport committed to relief the needs of the customer. Autonomous

electric cars will be in the center of business trends ruled by IT platforms, which will catalyze autonomous electric cars as regular transport. In the near future, innovation on services will take the use of autonomous electric cars to the next level.

The future driver or user profile of centennials will appreciate the value proposition of the autonomous electric car. There will not be need of finding parking, neither need to pay for it. An autonomous car will meet the user in the place and time that the user needs. It is estimated that as well as millennials, for centennials and next generations property is not in the first row of their needs, but the work to be done by the service is the most important thing when younger generations buy or rent a service. So, it seems that for future generations of drivers, those autonomous electric cars paid as a service, by minute, will be a reality.

**Acknowledgments** The authors would like to thank Center of Innovation and Design (CEID) of CETYS University Mexicali Campus for all facilities to perform the research and for providing the necessary resources to develop this project. Also, special thanks to the image illustrators Luis Esquivel, Alexa Macías, and Valeria Muñoz.

## References

1. Bauman, Z. (2000). *Liquid modernity*. Cambridge, UK: Polity Press.
2. Chesbrough, H. W. (2003). *Open innovation: The new imperative for creating and profiting from technology*. Boston: Harvard Business School Press.
3. Chesbrough, H. W. (2003). The era of open innovation. *MIT Sloan Management Review*, 44(3), 35–41.
4. Alexander, O., & Pigneur, Y. (2010). *Business model generation*. Hoboken, NJ: Wiley.
5. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles J3016\_201806. (2018). Retrieved February 5, 2019, from [https://www.sae.org/standards/content/j3016\\_201806/](https://www.sae.org/standards/content/j3016_201806/)
6. Raspberry Pi. (2019). Retrieved February 5, 2019, from <https://www.raspberrypi.org/>
7. Zini 1880. (2019). Retrieved February 5, 2019, from <https://zareason.com/zini-1880.html>
8. Al-Muteb, K., Faisal, M., Emaduddin, M., et al. (2016). An autonomous stereovision-based navigation system (ASNS) for mobile robots. *Intelligent Service Robotics*, 9, 187. <https://doi.org/10.1007/s11370-016-0194-5>.
9. What Is GPS? (2019). Retrieved February 5, 2019, from <https://www.gps.gov/systems/gps/>
10. BeiDou Navigation Satellite System. (2019). Retrieved February 6, 2019, from <http://en.beidou.gov.cn/>
11. Indian Regional Navigation Satellite System (IRNSS). (2019). Retrieved February 6, 2019, from <https://www.isro.gov.in/irnss-programme>
12. European Global Satellite-Based Navigation System. (2019). Retrieved February 6, 2019, from <https://www.gsa.europa.eu/european-gnss/galileo/galileo-european-global-satellite-based-navigation-system>
13. Graham, A. (2010). *Communications, radar and electronic warfare*. Hoboken: Wiley. Available from: ProQuest Ebook Central. [7 February 2019].
14. LIDAR vs RADAR Comparison. Which System is Better for Automotive? (2018). Retrieved February 7, 2019, from <https://www.archer-soft.com/en/blog/lidar-vs-radar-comparison-which-system-better-automotive>
15. Winner, H. (2016). Automotive RADAR. In H. Winner, S. Hakuli, F. Lotz, & C. Singer (Eds.), *Handbook of driver assistance systems*. Cham: Springer.

16. Mobile Industrial Robots. (2019). Retrieved February 11, 2019, from <http://www.jacobsonconstruction.com/projects/dabc-asrs-expansion-warehouse-remodel/>
17. Ekren, B. Y., & Heragu, S. S. (2012). A new technology for unit-load automated storage system: Autonomous vehicle storage and retrieval system. In R. Manzini (Ed.), *Warehousing in the global supply chain*. London: Springer. [https://doi.org/10.1007/978-1-4471-2274-6\\_12](https://doi.org/10.1007/978-1-4471-2274-6_12).
18. Kuo, P.-H., et al. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, 31(10), 2332–2346. <https://doi.org/10.1016/j.apm.2006.09.011>.
19. Waymo unveils self-driving taxi service in Arizona for paying customers. (2018). Retrieved February 11, 2019, from <https://www.reuters.com/article/us-waymo-selfdriving-focus/waymo-unveils-self-driving-taxi-service-in-arizona-for-paying-customers-idUSKBN1O41M2>
20. iRobot. (2019). Retrieved February 11, 2019, from <https://store.irobot.com/default/home>
21. Özgüner, U., Acarman, T., & Redmill, K. (2011). *Autonomous ground vehicles* (pp. 69–106). Boston: Artech House.
22. Weitkamp, C. (2005). *Lidar* (pp. 3–4). New York, NY: Springer.
23. Caltagirone, L., Scheidegger, S., Svensson, L., & Wahde, M. (2017). Fast LIDAR-based road detection using fully convolutional neural networks. In *2017 IEEE Intelligent Vehicles Symposium (IV)*.
24. Velodyne VLP-16. (2019). Retrieved February 27, 2019, from <https://velodynelidar.com/vlp-16.html/>
25. Velodyne HDL-64E. (2019). Retrieved February 27, 2019, from <https://velodynelidar.com/hdl-64e.html/>
26. Renishaw plc. Optical Encoders and LiDAR Scanning. (2019). Retrieved February 27, 2019, from <https://www.renishaw.it/it/optical-encoders-and-lidar-scanning%2D%2D39244/>
27. YeeFen Lim, H. (2018). *Autonomous vehicles and the law: Technology, algorithms and ethics* (p. 28). Edward Elgar Publishing.
28. InnovizOne. (2019). Retrieved February 27, 2019, from <https://innoviz.tech/innovizone/>
29. Kinect Sensor. (2019). Retrieved February 27, 2019, from Amir, S., Waqar, A., Siddiqui, M. A., et al. (2017). Kinect controlled UGV. *Wireless Personal Communications* 95, 631. <https://doi.org/10.1007/s11277-016-3915-3>.
30. Giori, C., & Fascinari, M. (2013). *Kinect in motion* (pp. 9–10). Birmingham, UK: Packt Pub..
31. Bernini, N., Bertozzi, M., Castangia, L., Patander, M., & Sabbatelli, M. (2014). Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.
32. Thrun, S., Burgard, W., & Fox, D. (2006). *Probabilistic robotics* (p. 221). Cambridge, MA: MIT Press.
33. Siciliano, B., & Khatib, O. (2008). *Springer handbook of robotics* (p. 857). Berlin: Springer.
34. Li, Z., Zhu, Q., & Gold, C. (2004). *Digital terrain modeling* (p. 7). Boca Raton: CRC Press.
35. Mach, R., & Petschek, P. (2007). *Visualization of digital terrain and landscape data* (p. 38). Berlin: Springer.
36. Hernandez-Aceituno, J., Arnay, R., Toledo, J., & Acosta, L. (2016). Using Kinect on an autonomous vehicle for outdoors obstacle detection. *IEEE Sensors Journal*, 16(10), 3603–3610.
37. Wedel, A., & Cremers, D. (2011). *Stereo scene flow for 3D motion analysis* (p. 89). Springer.
38. Plemenos, D., & Miaoulis, G. (2013). *Intelligent computer graphics 2012* (pp. 243–263). Berlin: Springer.
39. Schaub, A. (2017). *Robust perception from optical sensors for reactive behaviors in autonomous robotic vehicles* (p. 161). Springer.
40. Jensen, M., Philipsen, M., Mogellose, A., Moeslund, T., & Trivedi, M. (2016). Vision for looking at traffic lights: Issues, survey, and perspectives. *IEEE Transactions on Intelligent Transportation Systems*, 17(7), 1800–1815.

41. Mogelmoose, A., Trivedi, M., & Moeslund, T. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Transactions on Intelligent Transportation Systems*, 13(4), 1484–1497.
42. Trepagnier, P., Nagel, J., & McVay Kinney, P. Navigation and control system for autonomous vehicles. US Patent 8,050,863 B2.
43. Cox, I., & Wilfong, G. (1990). *Autonomous robot vehicles*. New York, NY: Springer.
44. Jiang, X., Hornegger, J., & Koch, R. (2014). *Pattern recognition* (p. 4). Cham: Springer.
45. Dhiman, N. K., Deodhare, D., & Khemani, D. (2015). Where am I? Creating Spatial awareness in unmanned ground robots using SLAM: A survey. *Sadhana Academy Proceedings in Engineering Sciences*, 40(5), 1385–1433. <https://doi.org/10.1007/s12046-015-0402-6>.
46. Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., Reid, I., & Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust perception age. *IEEE Transactions on Robotics*, 32(6), 1309–1332. Retrieved from <https://ieeexplore.ieee.org/document/7747236>.
47. Siegwart, R., & Nourbakhsh, I. R. (2004). *Introduction to autonomous mobile robots*. Cambridge, MA: MIT Press.
48. Puente, I., González-Jorge, H., Martínez-Sánchez, J., & Arias, P. (2013). Review of mobile mapping and surveying technologies. *Measurement*, 46(7), 2127–2145. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0263224113000730>.
49. Gonzalez-Jorge, H., Rodríguez-González, P., Martínez-Sánchez, J., González-Aguilera, D., Arias, P., Gesto, M., & Díaz-Vilariño, L. (2015). Metrological comparison between Kinect I and Kinect II sensors. *Measurement*, 70, 21–26.
50. Fankhauser, P., Bloesch, M., Rodriguez, D., Kaestner, R., Hutter, M., & Siegwart, R. (2015, July). Kinect v2 for mobile robot navigation: Evaluation and modeling. In *2015 International Conference on Advanced Robotics (ICAR), Istanbul*, pp. 388–394. Retrieved from <https://ieeexplore.ieee.org/document/7251485>
51. Sell, J., & O'Connor, P. (2014). The Xbox one system on a chip and Kinect sensor. *IEEE Micro*, 34(2), 44–53. Retrieved from <https://ieeexplore.ieee.org/document/6756701>.
52. Durrant-Whyte, H., & Bailey, T. (2016). Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *IEEE Robotics and Automation Magazine*, 13(2), 99–110. Retrieved from <https://ieeexplore.ieee.org/document/1638022>.
53. Schadler, M., Stückler, J., & Behnke, S. (2014). Rough terrain 3D mapping and navigation using a continuously rotating 2D laser scanner. *Künstliche Intelligenz*, 28(2), 93–99. <https://doi.org/10.1007/s13218-014-0301-8>.
54. Beul, M., Krombach, N., Zhong, Y., Droeschel, D., Nieuwenhuisen, M., & Behnke, S. (2015, July). A high-performance MAV for autonomous navigation in complex 3D environments. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS), Denver, CO*. <https://ieeexplore.ieee.org/document/7152417>
55. Gupta, S., Davidson, J., Levine, S., Sukthankar, R., & Malik, J. (2017, November). Cognitive mapping and planning for visual navigation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI*. Retrieved from <https://ieeexplore.ieee.org/document/8100252>
56. Lacaze, A., Moscovitz, Y., DeClaris, N., & Murphy, K. Path planning for autonomous vehicles driving over rough terrain. In *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell.*
57. Ferguson, D., & Stentz, A. *The Field D\* algorithm for improved path planning and replanning in uniform and non-uniform cost environments*. Technical Report CMU-TR-RI-05-19, Carnegie Mellon University.
58. Básaca-Preciado, L. C., Orozco-García, N. A., & Terrazas-Gaynor, J. M., et al. (2018). *Intelligent transportation scheme for autonomous vehicle in smart campus*. IEEE, pp. 3193–3199.
59. Martínez-Austria, P. F., Bandala, E. R., & Patiño-Gómez, C. (2016). Temperature and heat wave trends in northwest Mexico. *Physics and Chemistry of the Earth, Parts A/B/C*, 91, 20–26.

60. Åström, D. O., Bertil, F., & Joacim, R. (2011). Heat wave impact on morbidity and mortality in the elderly population: A review of recent studies. *Maturitas*, *69*, 99–105.
61. Básaca-Preciado, L. C., et al. (2014). Optical 3D laser measurement system for navigation of autonomous mobile robot. *Optics and Laser in Engineering*, *54*, 159–169. <https://doi.org/10.1016/j.optlaseng.2013.08.005>.
62. Lucas, H. C., Jr., et al. (2009). Disruptive technology: How Kodak missed the digital photography revolution. *Journal of Strategic Information Systems*, *18*, 46–55.
63. Resident population in the United States in 2017, Statista. (2018). *The Statistics Portal*. Retrieved from January 25, 2019, from <https://www.statista.com/statistics/797321/us-population-by-generation/>
64. Díaz Caravantes, R. E., Castro Luque, A. L., & Aranda Gallegos, P. (2014). Mortality by excessive natural heat in Northwest Mexico: Social conditions associated with this cause of death. *Front Norte*, *26*, 155–177.