



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**CRET - CanBus Reverse
Engineering Toolkit**



Presentado por Adrián Marcos Batlle
en Universidad de Burgos — 17 de enero
de 2019

Tutor: Álvaro Arnaiz-González



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



D. Álar Arnaz-González, profesor del departamento de Ingeniería Civil, área de Lenguajes y Sistemas Informáticos.

Expone:

Que el alumno D. Adrián Marcos Batlle, con DNI 71310384B, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado CRET - CanBus Reverse Engineering Toolkit.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 17 de enero de 2019

Vº. Bº. del Tutor:

Vº. Bº. del co-tutor:

D. nombre tutor

D. nombre co-tutor

Resumen

El bus CAN (CAN-Bus - Controlled Area Network) es un protocolo de comunicación utilizado en los desarrollos de multitud de sectores críticos como la industria, la automoción y la aviación entre otros, para la comunicación entre los componentes internos que forman la infraestructura desarrollada.

Los datos que fluyen por dicho bus son propiedad de cada uno de los fabricantes a pesar de ser un protocolo libre. El desarrollo de esta herramienta viene motivado a realizar un análisis de la información que fluye por estos buses, así como su clasificación y monitorización en tiempo real.

Para dicho objetivo, se ha desarrollado tanto una parte de software (para el análisis y clasificación de los datos), como una parte de hardware para conectarse a dicho bus y poder acceder a los datos que fluyen a través del mismo.

Descriptores

CAN-Bus, Automoción, Industria, Aviónica, Ingeniería inversas, Análisis de protocolos.

Abstract

A **brief** presentation of the topic addressed in the project.

Keywords

Can-Bus, Automotive, Industry, Avionics, Reversing, Protocol analysis.

Índice general

Índice general	III
Índice de figuras	V
Índice de tablas	VI
Introducción	1
1.1. Estructura de la memoria	1
1.2. Materiales adjuntos	2
Objetivos del proyecto	3
2.1. Objetivos generales	3
2.2. Objetivos técnicos	3
2.3. Objetivos personales	3
Conceptos teóricos	5
3.1. Secciones	6
3.2. Referencias	6
3.3. Imágenes	6
3.4. Listas de items	7
3.5. Tablas	8
Técnicas y herramientas	9
Aspectos relevantes del desarrollo del proyecto	11
Trabajos relacionados	13

Conclusiones y Líneas de trabajo futuras	15
---	-----------

Índice de figuras

3.1. Autómata para una expresión vacía	7
--	---

Índice de tablas

3.1. Herramientas y tecnologías utilizadas en cada parte del proyecto	8
---	---

Introducción

Todos los vehículos que utilizamos en el día a día, maquinaria utilizada en las empresas, el sector náutico o de la aviación utilizan el bus CAN para la inter-conexión de los componentes electrónicos que hacen funcionar dichas máquinas.

Durante los años se han desarrollado nuevos protocolos, todos ellos basados en (habiéndose incrementado, por ejemplo, la velocidad en los buses actuales), pero utilizando las bases del protocolo que en su primer desarrollo.

El estándar CAN (CITA) es

El estándar del bus CAN únicamente hace referencia a las dos primeras capas del protocolo, la capa física, y la capa de enlace de datos (siguiendo el modelo OSI).

A través del uso de esta herramienta, sería posible identificar y clasificar los datos que los distintos elementos del vehículo analizado comparten entre ellos, para su funcionamiento. De esta manera, por ejemplo, si necesitásemos realizar una aplicación para la monitorización de un vehículo, no sería necesario introducir nuevos sensores (para la velocidad, las revoluciones del motor, el GPS), sino que estos datos serían extraídos del bus CAN, ahorrando costes y posibles problemas.

1.1. Estructura de la memoria

Introducción: Descripción breve sobre el proyecto, motivación por la que se ha realizado y soluciones propuestas. Estructura de la memoria y listado de materiales adjuntos proporcionados.

Objetivos del proyecto: Exposición de los objetivos, clasificados en objetivos generales, objetivos técnicos y objetivos personales.

Conceptos teóricos: Conceptos básicos y necesarios para entender el propósito del proyecto.

Técnicas y herramientas: Metodologías y herramientas utilizadas durante el desarrollo del proyecto.

Trabajos relacionados: Aplicaciones, proyectos y empresas que ofrecen soluciones en el mismo campo que el estudiado.

Conclusiones y líneas de trabajo futuras: Conclusiones a las que se ha llegado tras la realización del proyecto, así como mejoras y futuro desarrollo de la aplicación.

1.2. Materiales adjuntos

Los materiales adjuntos a la memoria son los siguientes:

- Aplicación desarrollada en Java: CRET.
- Fotos del hardware desarrollado.
- Esquemas del hardware desarrollado.
- JavaDoc.

Además, los siguientes recursos están accesibles a través de internet:

- Repositorio del proyecto TODO—.

Objetivos del proyecto

A continuación se definen los objetivos del proyecto realizado, estructurados en tres secciones:

2.1. Objetivos generales

- Desarrollar una aplicación para el análisis y monitorización de los datos que fluyen por el bus CAN.
- Desarrollo de un hardware libre el cual permita conectarse a dicho bus de datos y monitorizar distintas velocidades de forma simultanea.

2.2. Objetivos técnicos

- Desarrollar un *hardware* propio desde 0 siguiendo la metodología para el desarrollo del mismo, así como su producción y montaje.
- Desarrollar una aplicación en Java y JavaFX.
- Aplicar la arquitectura MVP (*Model-View-Presenter*) en el desarrollo de la aplicación.
- Utilizar Zenhub (basado en el método Kanban) para realizar un seguimiento y gestión del proyecto.
- Utilizar Git (en la plataforma GitHub) para realizar un control de versiones de software.
- Utilizar librerías para la recolección de datos del bus CAN.

2.3. Objetivos personales

- Profundizar en el conocimiento de hardware y en el desarrollo del mismo.
- Adquirir conocimiento sobre el funcionamiento del bus CAN en distintos escenarios.
- Profundizar en el conocimiento del análisis de datos y monitorización en tiempo real.

Conceptos teóricos

Las partes del proyecto con mayor desconocimiento y complejidad están enfocadas principalmente en el funcionamiento del bus CAN y en el desarrollo del hardware, el cual requiere de unos conocimientos básicos y unas metodologías específicas las cuales serán detalladas a continuación:

Bus CAN

El bus CAN (*Controller Area Network*) es un protocolo desarrollado para la comunicación entre los distintos micro-controladores y dispositivos que son necesarios para el funcionamiento de un vehículo. Este protocolo no necesita de un host principal, sino que sigue una topología de tipo "bus".

Es utilizado en multitud de escenarios como la aviación, la navegación, la automatización industrial, instrumentos médicos, maquinaria pesada y ascensores entre otros.

Este protocolo está basado en el uso de mensajes para el intercambio de información entre los distintos dispositivos que lo componen.

Nodo: Cada uno de los dispositivos físicos que están conectados a la red CAN. Al menos es necesario que existan dos nodos conectados a la red para que se produzca una comunicación.

Cada uno de los nodos es capaz de enviar y recibir mensajes, pero no simultáneamente.

FOTO DE UN SOLO NODO.

Frame: Cada uno de los mensajes enviados a través de la red CAN.

ID:

Length:

Data:

Siguiendo el modelo OSI, el estándar bus CAN especifica únicamente las dos primeras capas:

Capa física:

Capa de enlace de datos:

CanBus Esquemas can Explicación ID Explicación Velocidad Explicación hardware Explicación señales low-level Explicación capas del canbus Explicación modo activo-listenonly

Explicación conceptos de hardware: footprint scheme gerber files

3.1. Secciones

Las secciones se incluyen con el comando section.

Subsecciones

Además de secciones tenemos subsecciones.

Subsubsecciones

Y subsecciones.

3.2. Referencias

Las referencias se incluyen en el texto usando cite [?]. Para citar webs, artículos o libros [?].

3.3. Imágenes

Se pueden incluir imágenes con los comandos standard de L^AT_EX, pero esta plantilla dispone de comandos propios como por ejemplo el siguiente:



Figura 3.1: Autómata para una expresión vacía

3.4. Listas de items

Existen tres posibilidades:

- primer item.
- segundo item.

1. primer item.

Herramientas	App	AngularJS	API REST	BD	Memoria
HTML5		X			
CSS3		X			
BOOTSTRAP		X			
JavaScript		X			
AngularJS		X			
Bower		X			
PHP			X		
Karma + Jasmine		X			
Slim framework			X		
Idiorm			X		
Composer			X		
JSON		X	X		
PhpStorm		X	X		
MySQL				X	
PhpMyAdmin				X	
Git + BitBucket		X	X	X	X
MikTeX					X
TeXMaker					X
Astah					X
Balsamiq Mockups		X			
VersionOne		X	X	X	X

Tabla 3.1: Herramientas y tecnologías utilizadas en cada parte del proyecto

2. segundo item.

Primer item más información sobre el primer item.

Segundo item más información sobre el segundo item.

■

3.5. Tablas

Igualmente se pueden usar los comandos específicos de \LaTeX o bien usar alguno de los comandos de la plantilla.

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas. No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

POMODORO

Patrones de diseño: MVC

Esquema global del programa, con input de can, etc Control de versiones con GIT Gestión del proyecto con zenhub Entorno de desarrollo eclipse + scenebuilder latex javafx

Librerías: usbtin serial ports

Hardware: kicad

Aspectos relevantes del desarrollo del proyecto

Inicio del proyecto

Motivaciones ,etc logo

Metodologías

Formación

Desarrollo del software

Desarrollo de la app

Desarrollo del hardware

Trabajos relacionados

Cada fabricante dispone de sus propias herramientas (al igual que muchos de ellos disponen de su propio hardware). Estas herramientas, como se ha mencionado en la introducción, tienen un coste muy elevado en la mayoría de los casos.

Las principales ventajas de este proyecto sobre los otros mencionados anteriormente son:

El hardware desarrollado puede ser producido por cualquier persona, ya que los esquemas y los ficheros necesarios para su producción son open source.

Aplicación multiplataforma, compatible tanto con sistemas Windows como Linux.

No es necesaria la instalación de drivers en el equipo en el que se va a utilizar la herramienta.

Es la única herramienta libre que permite "graficar" los datos que fluyen por el bus CAN, así como su identificación y posterior monitorización.

Las principales desventajas son:

Actualmente solo funciona con un hardware en concreto. Como se menciona en la sección –Lineas futuras de trabajo–, el desarrollo de este proyecto continuaría con la incorporación de un módulo para el soporte de los drivers SocketCAN en sistemas linux.

Conclusiones y Líneas de trabajo futuras

Conclusiones:

Líneas de trabajo futuras:

Es interesante proseguir con el desarrollo de la herramienta y adaptarla para su utilización con los módulos de Linux SocketCAN.

De esta manera no es un requisito necesario el uso de un hardware en concreto, ya que SocketCAN haría de intermediario entre cualquier software soportado y la aplicación.

Además sería posible la creación y utilización de interfaces virtuales dentro del equipo, para realizar diversas pruebas.