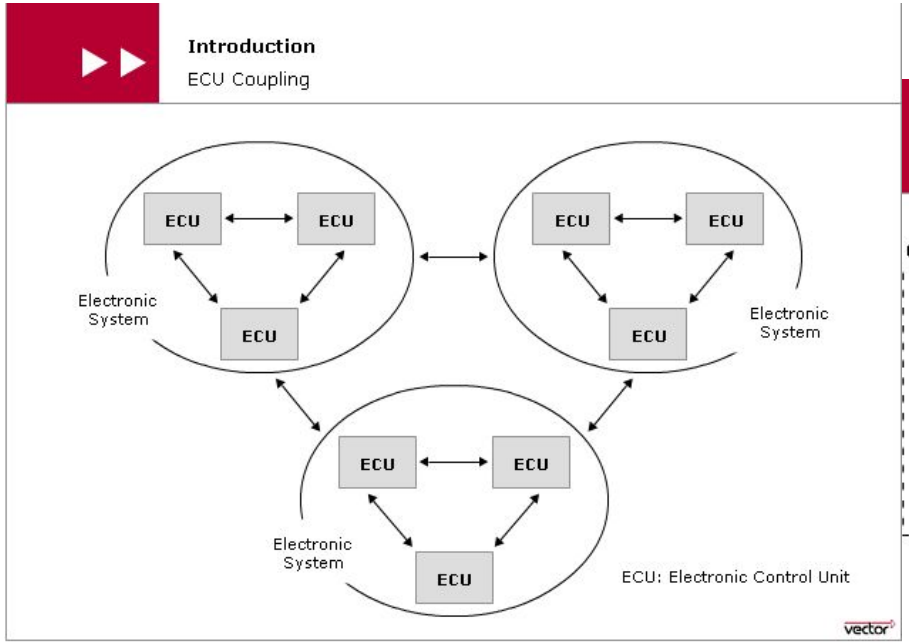
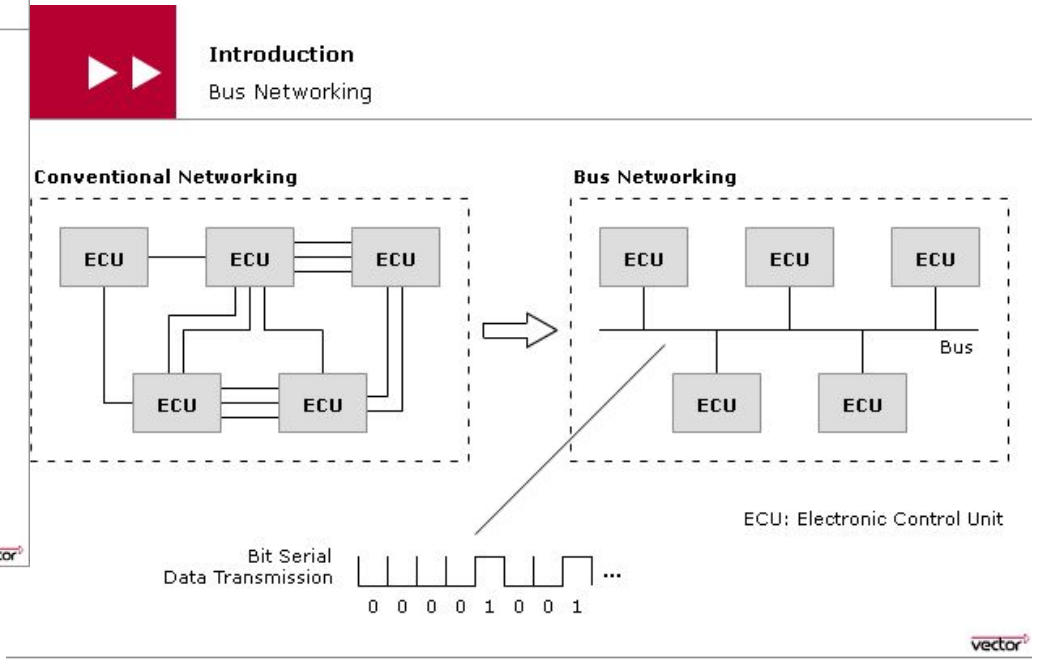
CAN educational prototype development

Expo Ciencias Nacional 2016

UPAEP

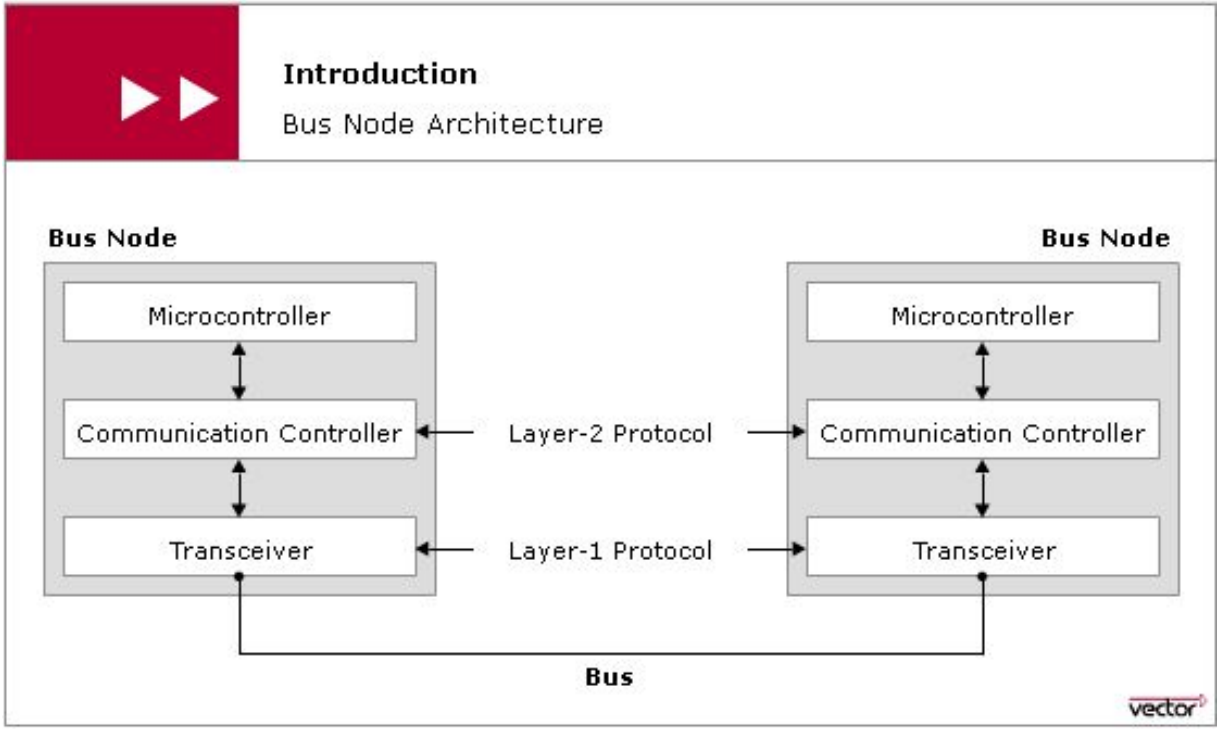
BUS System


Introduction
ECU Coupling
ECU
ECU
ECU
Electronic System
ECU
ECU
ECU
Electronic System
ECU
ECU
ECU
Electronic System
ECU
ECU: Electronic Control Unit
vector

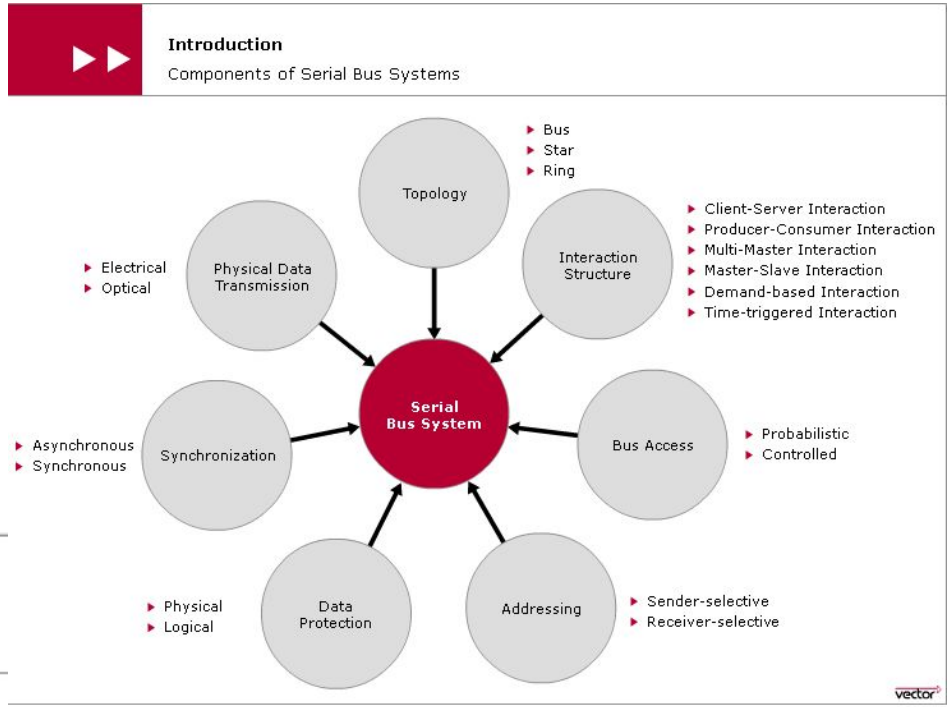

Introduction
Bus Networking
Conventional Networking
Bus Networking
ECU
ECU
ECU
ECU
ECU
ECU
ECU
ECU
ECU
ECU
Bus
ECU: Electronic Control Unit
Bit Serial
Data Transmission
0 0 0 0 1 0 0 1
vector

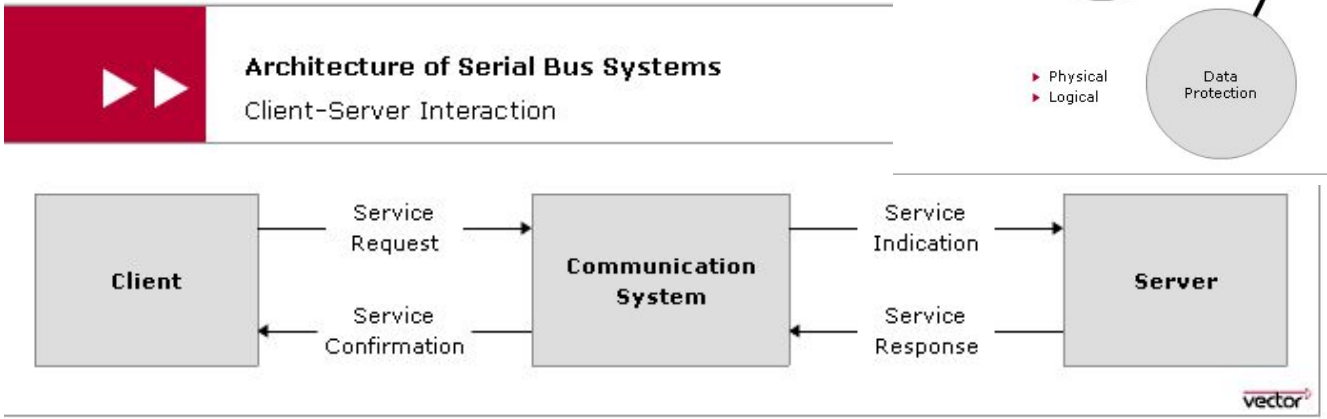
Bus System architecture

- Three layer model

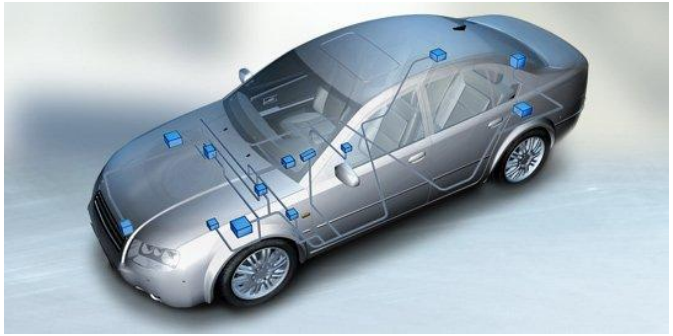


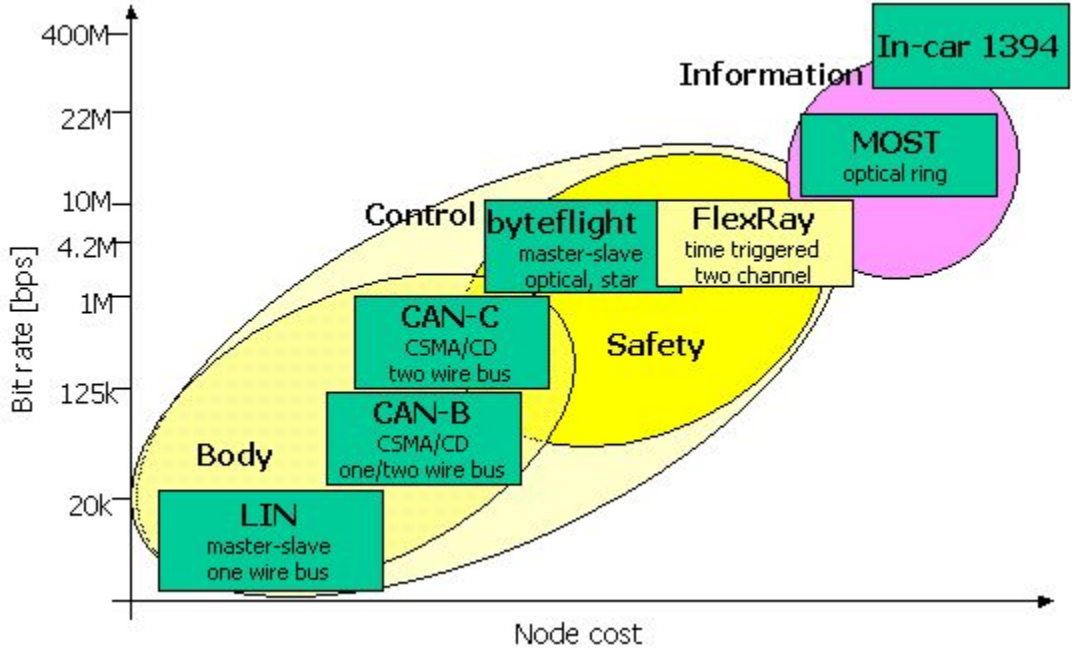
Bus system architecture





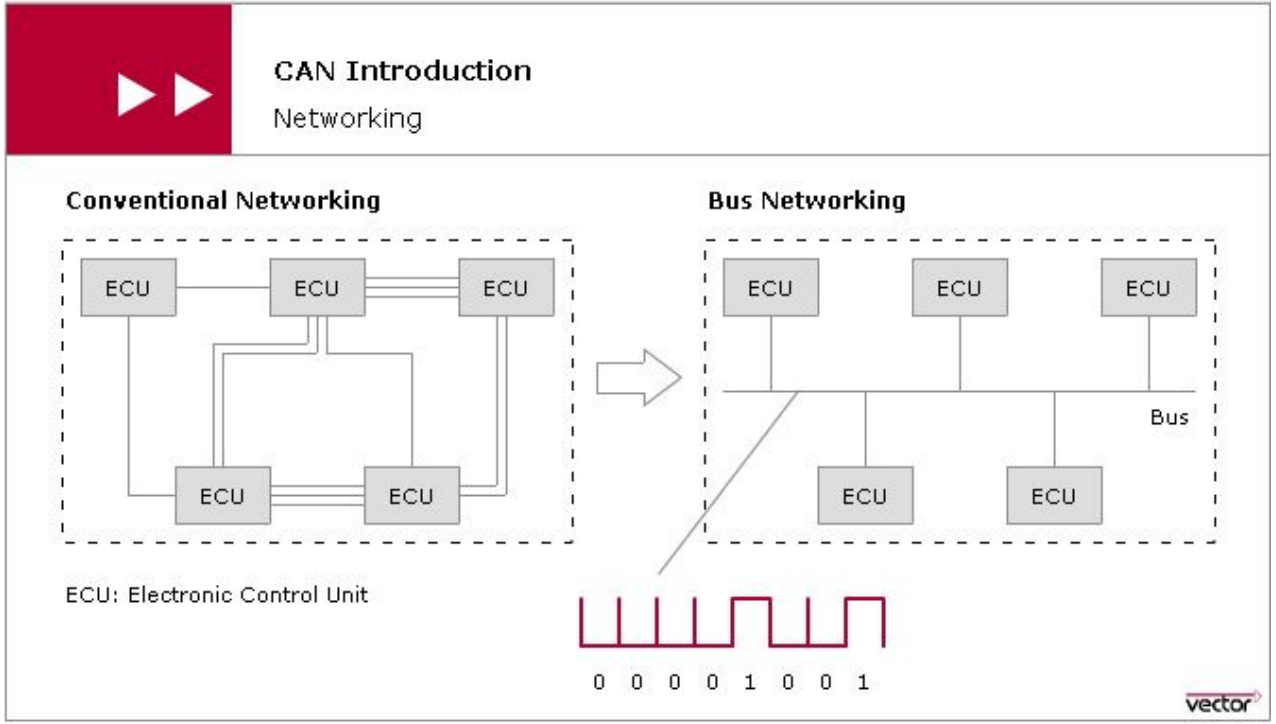
Bus systems



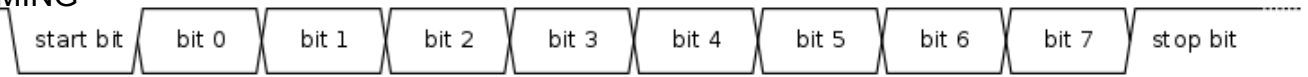

400M
22M
10M
4.2M
1M
125k
20k
Bit rate [bps]
Node cost
In-car 1394
Information
MOST
optical ring
Control
byteflight
master-slave
optical, star
FlexRay
time triggered
two channel
CAN-C
CSMA/CD
two wire bus
Safety
CAN-B
CSMA/CD
one/two wire bus
Body
LIN
master-slave
one wire bus


lin
Ethernet
MOST
ABC
CAN
FlexRay

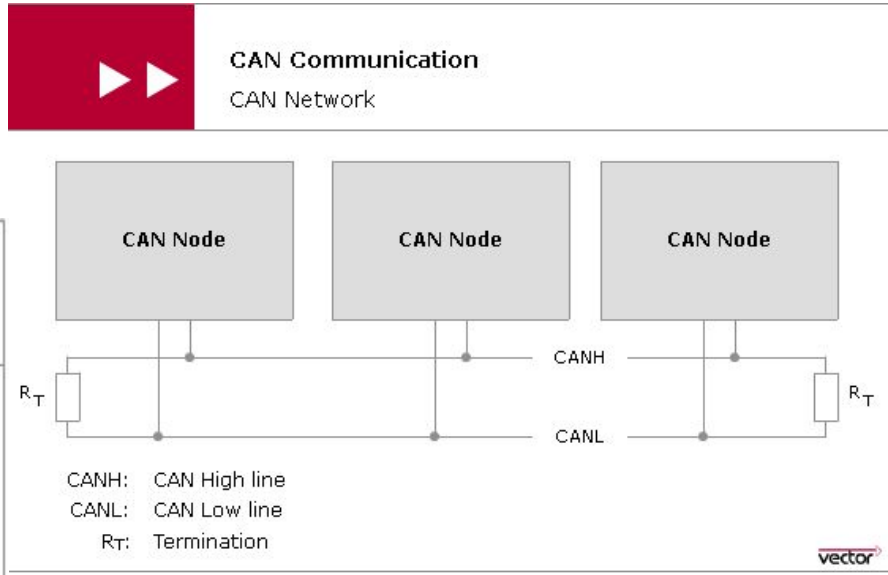
CAN Basics

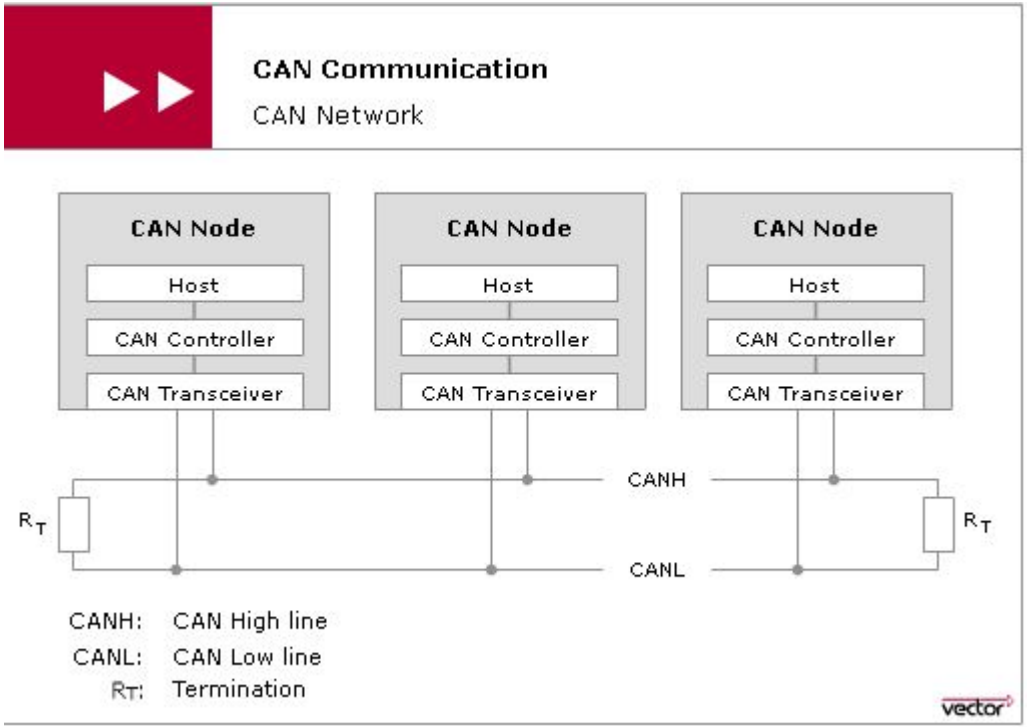


DATA FRAMING

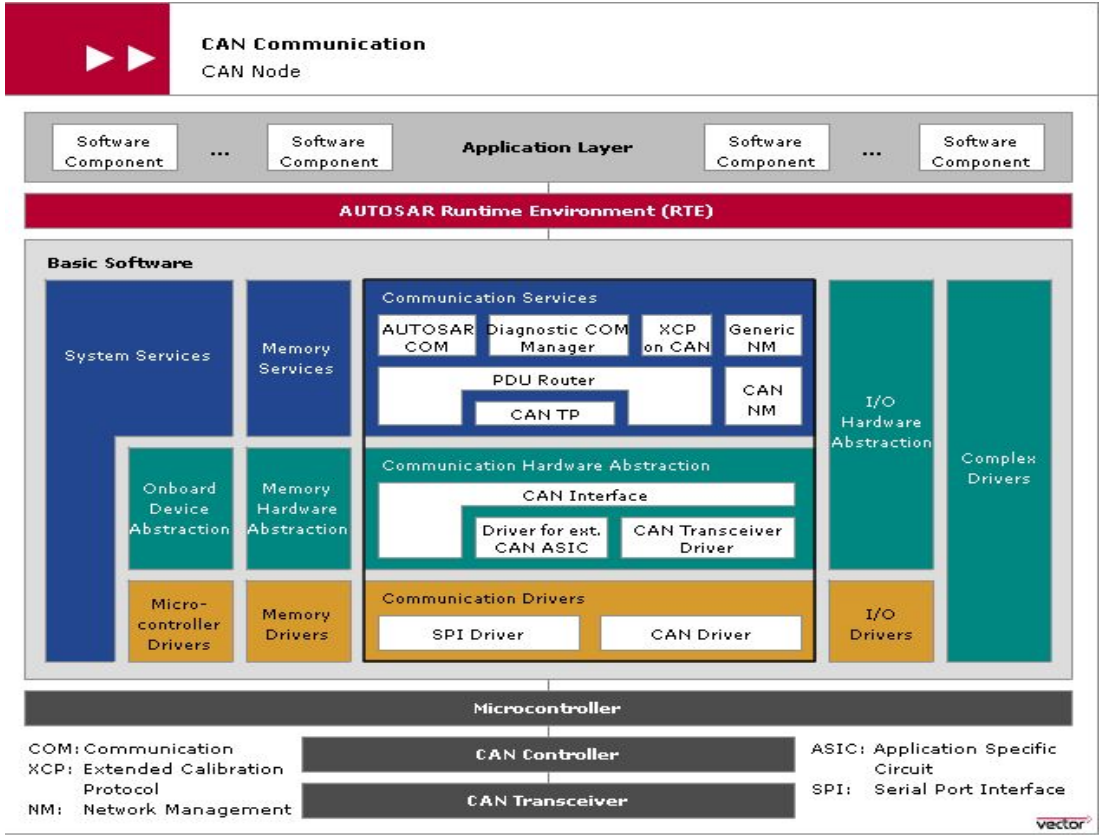


CAN Network

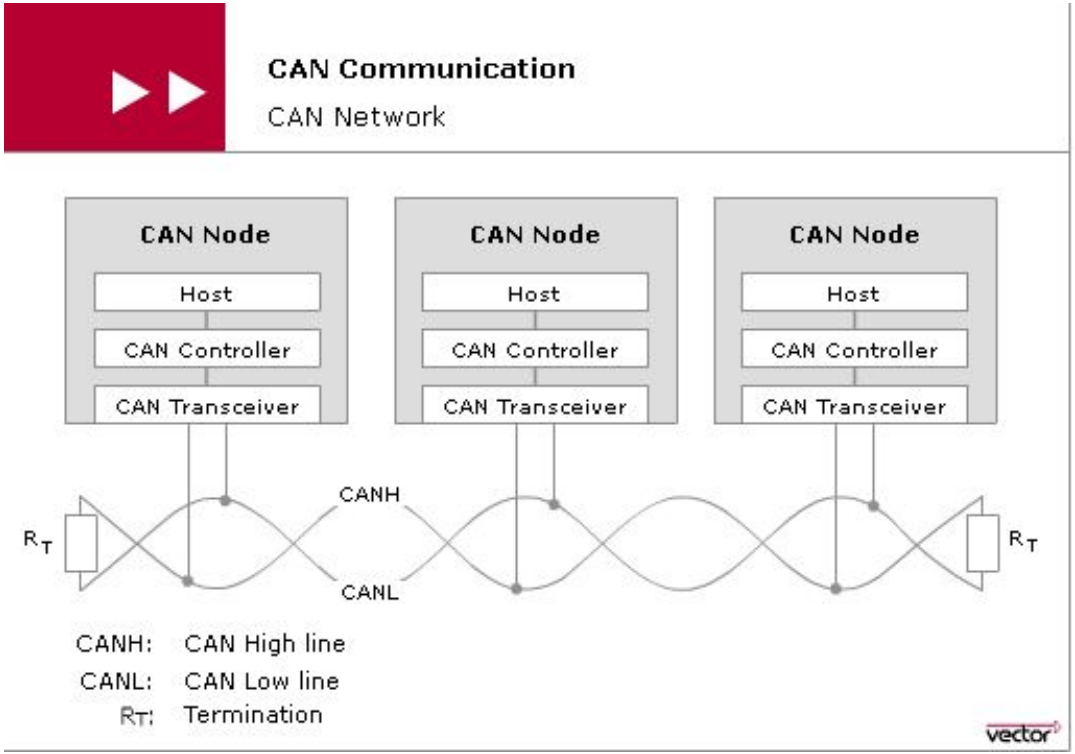

CAN Communication
CAN Network
CAN Node
CAN Node
CAN Node
CANH
CANL
R_T
R_T
CANH: CAN High line
CANL: CAN Low line
RT: Termination
vector


CAN Communication
CAN Network
CAN Node
Host
CAN Controller
CAN Transceiver
CAN Node
Host
CAN Controller
CAN Transceiver
CAN Node
Host
CAN Controller
CAN Transceiver
CANH
CANL
R_T
R_T
CANH: CAN High line
CANL: CAN Low line
RT: Termination
vector

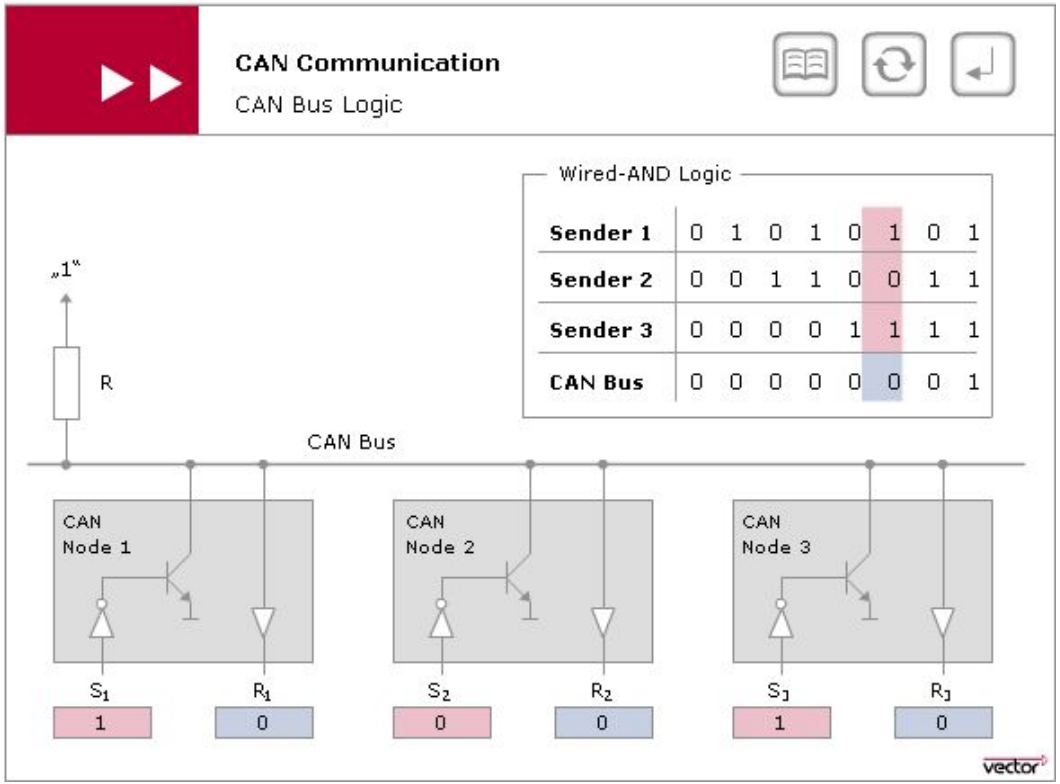
CAN Node


CAN Communication
CAN Node
Software Component
...
Software Component
Application Layer
Software Component
...
Software Component
AUTOSAR Runtime Environment (RTE)
Basic Software
System Services
Memory Services
Communication Services
AUTOSAR COM
Diagnostic COM Manager
XCP on CAN
Generic NM
PDU Router
CAN TP
CAN NM
I/O Hardware Abstraction
Complex Drivers
Onboard Device Abstraction
Memory Hardware Abstraction
Communication Hardware Abstraction
CAN Interface
Driver for ext. CAN ASIC
CAN Transceiver Driver
Micro-controller Drivers
Memory Drivers
Communication Drivers
SPI Driver
CAN Driver
I/O Drivers
Microcontroller
CAN Controller
CAN Transceiver
COM: Communication
XCP: Extended Calibration Protocol
NM: Network Management
ASIC: Application Specific Circuit
SPI: Serial Port Interface
vector

CAN Bus


CAN Communication
CAN Network
CAN Node
Host
CAN Controller
CAN Transceiver
CAN Node
Host
CAN Controller
CAN Transceiver
CAN Node
Host
CAN Controller
CAN Transceiver
CANH
CANL
R_T
R_T
CANH: CAN High line
CANL: CAN Low line
R_T: Termination
vector

CAN Logic


CAN Communication
CAN Bus Logic
Wired-AND Logic
Sender 1 0 1 0 1 0 1 0 1
Sender 2 0 0 1 1 0 0 1 1
Sender 3 0 0 0 0 1 1 1 1
CAN Bus 0 0 0 0 0 0 0 1
„1“
R
CAN Bus
CAN Node 1
CAN Node 2
CAN Node 3
S1
R1
S2
R2
S3
R3
1
0
0
0
1
0
vector

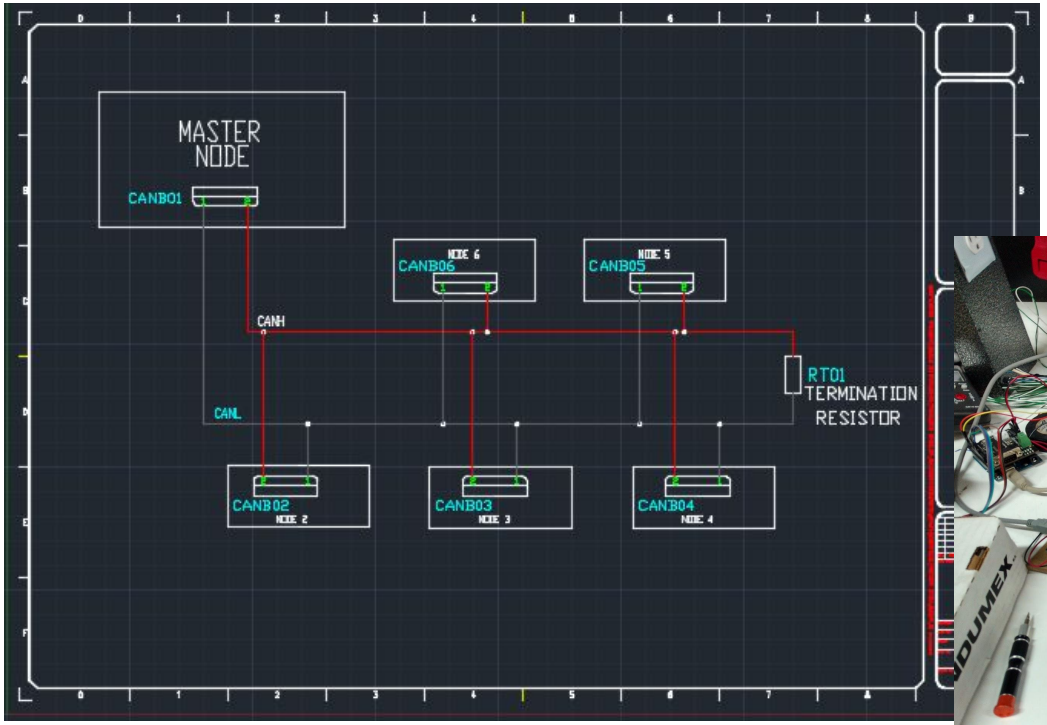
Motivation

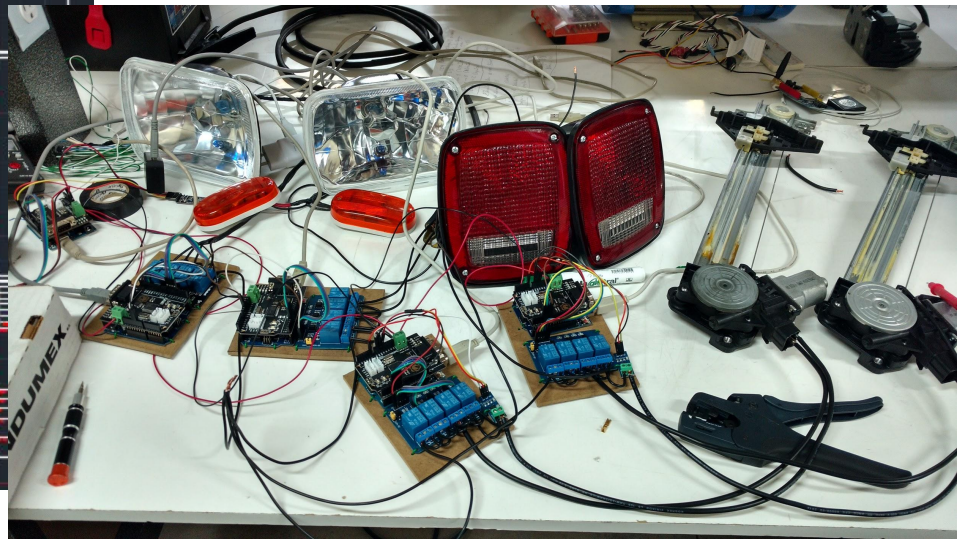
- Since the early developments in the 80´s to nearly autonomous car, all thanks to advanced **bus systems**.
- Customer security and satisfaction have also played a huge role in the development of this technology.
- Getting a better sense of the **network** and the **CAN bus** since early **automotive** education in a car can lead to major advancements in the **automotive** industry.

Objective

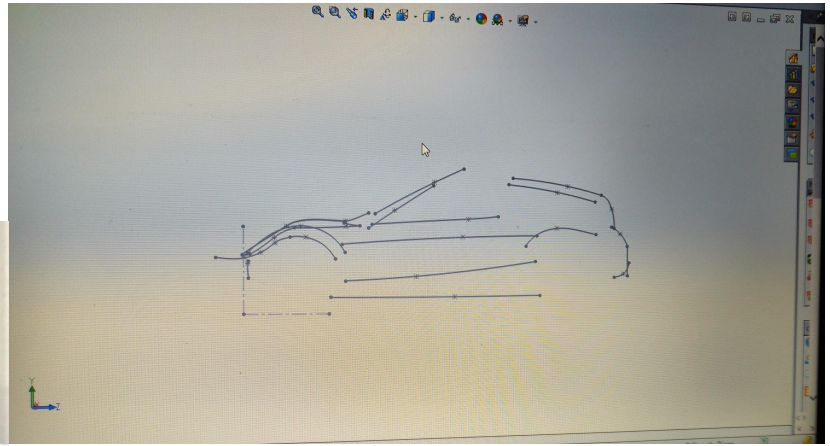
1. Study common **communication protocols** used in a car for exchange of **data.**
2. Once understood, we developed a **serial communication system** using **microcontrollers** and normal car components under the **CAN protocol**.
3. In addition to normal car components, new applications **modules** were introduced.
4. **Prototype** chassis was developed to present more visually the **network.**

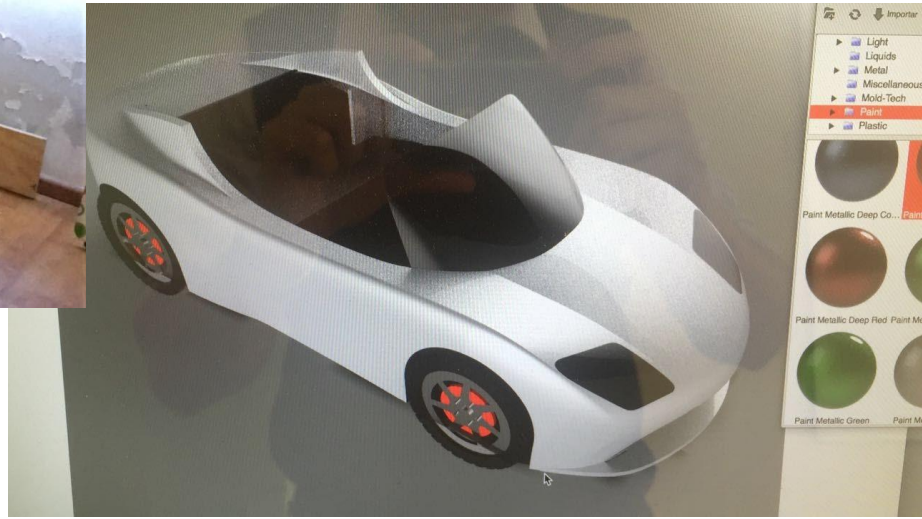
CAN prototype development





Prototype (Visual part)



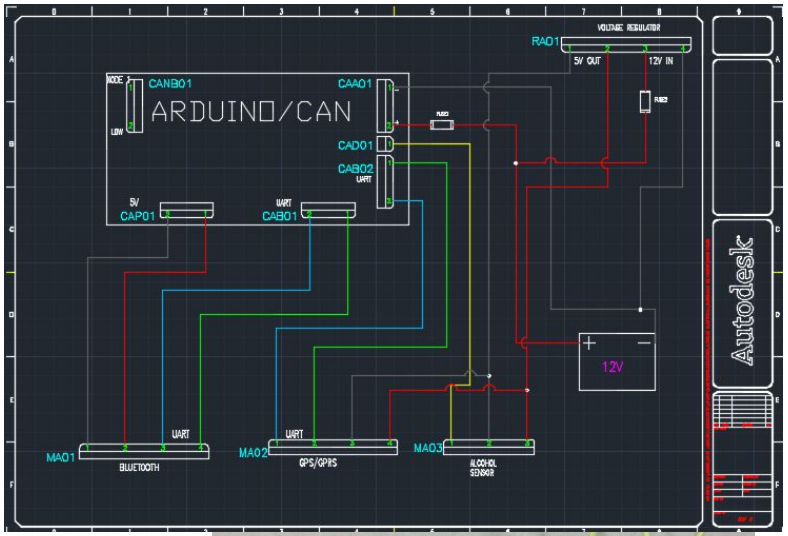


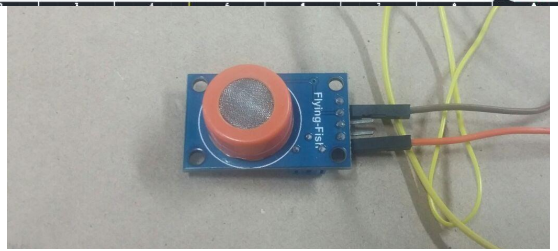
Benefits of prototype

- Educational and training module for learning the basics of CAN bus and automotive harness
- Interactive chassis with real and virtualized instrumentation

New additions/ better user experience

Security: Alcohol module





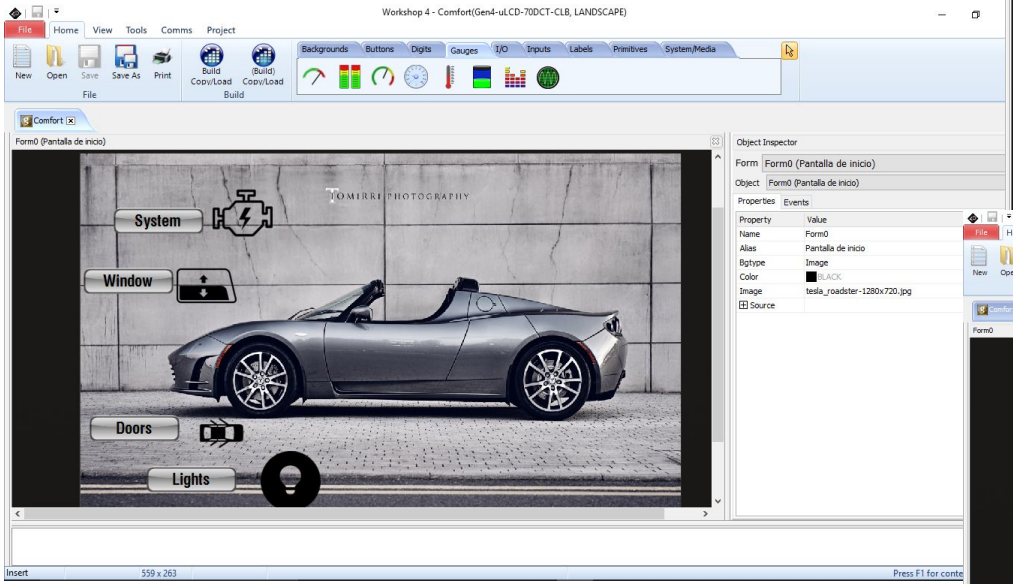
New additions/ better user experience

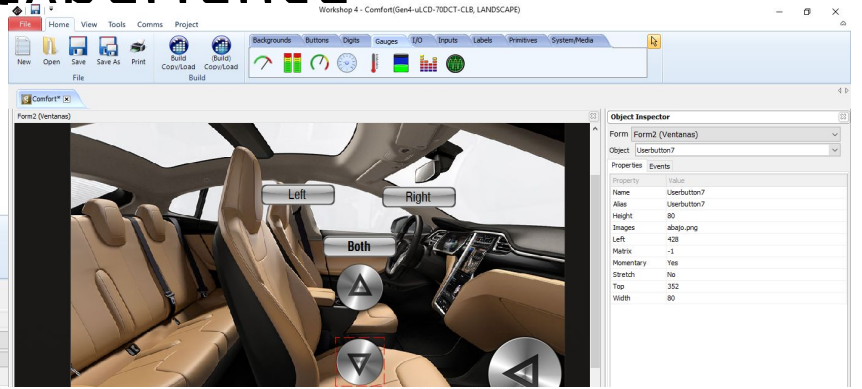
Comfort: Bluetooth control by app and RFID key

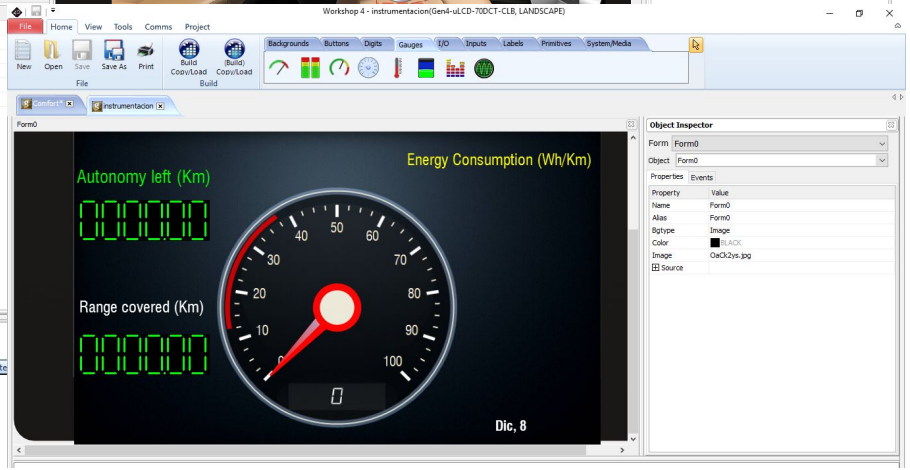
Touch displays

New additions/ better user experience

Comfort: Touch displays







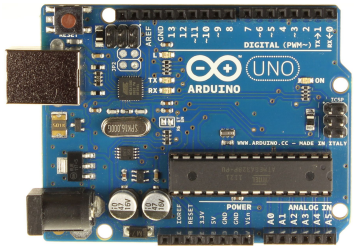
Why Arduino?

- Open source software (everyone can implement/change/see a code)
- Relatively cheap components and shields
- Free IDE
- Operates with low current
- Easy understanding for beginners to microcontrollers, control units and coding

Difference between microcontroller and PLC

Microcontroller

- Works with transistors (also maybe electronic relays)
- Doesn´t work as a stand alone controller
- Part of electronic circuit
- Easy implementation for working with CAN protocol (CAN Shield)
- Cheaper than a PLC



PLC

- Works with relays
- Works as a stand alone controlling device
- Number of inputs/outputs is larger
- No need for coding IDE and can be programmed in FBD (function block diagram) or ladder diagram
- Made for withstanding rough industrial environment
- Essentially for industrial automation
- No compatibility with shields
- Operates with high power

Contact

Manuel Rodriguez Marun - alex_marun93@hotmail.com

Kevin Aguilera Vazquez - kevin.aguiler21@gmail.com

Maria Fernanda Rivera Fernandez - mafer.rivf@gmail.com