

# CANalyzer CAN

## Agenda VectorAcademy

<b>Delivery Format:</b>	This Course is offered in Classroom or Remote Format
<b>Duration:</b>	Classroom: 1 day Remote: 7 hours
<b>Target Group:</b>	CAN ECU validation, software development and CAN architecture
<b>Prerequisites:</b>	Knowledge in CAN protocol, basic programming skills
<b>Goal:</b>	At the end of the training, the trainee will be able to analyze the CAN bus, report anomalies and send frames

---

**Evaluation:**

Validation of learning based on practical exercises with CANalyzer.

**Pedagogical, technical and supervisory resources:**

Course material is sent to each trainee. A link will be sent to each participant to install the CANoe software and use hardware interfaces. The training will be carried out in adapted rooms.

Competence of the trainer: 15 years of experience in training related to embedded developments, network architectures.

**Method of follow-up of the trainee:**

A sign-off sheet must be validated by the trainee. A first satisfaction questionnaire is planned at the end of the training.

---

**1. Introduction to CANalyzer**

- > Context for CANalyzer usage
- > Usage of CANalyzer for CAN ECU analysis

**2. Measurement and Analysis**

- > Configuration of the "Measurement Setup" window, Filter
- > Introduction to analysis windows
- > Traces or message traffic, statistics and signal analysis
- > Import and export of data, various data formats
- > Panels and analysis

**3. Data Recording**

- > Recording of data traffic, format of recording files
- > Using trigger conditions

**4. Offline Analysis**

- > Offline mode analysis of recorded data
- > Condition in offline execution

# CANalyzer CAN

## Agenda VectorAcademy

### 5. Functional Blocks

- > "Interactive Block Generator" to send frames
- > Replay Block

### 6. Introduction to CAPL

- > Motivation for CAPL to automate analysis
- > Introduction to CAPL and the development environment
- > Event procedures, database access

### 7. CAPL syntax and functions

- > Data type and Arithmetic and logical operations
- > Evaluation of a CAN message, control of the sending of a message,
- > Time-out management
- > Functions, libraries...