
Workshop:
Vehicle Networks
CanBusHack

Installation

- For those of you who have Windows PCs (10 PCs Needed!)
  - Please download (or grab thumb drive) a copy of Vehicle Spy from my Dropbox https://dl.dropbox.com/u/6645572/vspy3_install.zip
  - License Files will be Contained in the Install
  - If you don't trust me you can get it from the source, www.intrepidcs.com/main/updates. This ZIP file has a password, I will need to type it in for you.

Please organize yourselves

- Due to not having access to enough CAN <-> USB hardware devices, we will need to share
- Please arrange yourselves so that 3 people can Share one hardware!
- If you don’t want to participate in the activities, please move so that other can.
- If you do want to participate but there isn’t enough room, please sit by me.. I have room for two more.

CanBusHack

Outline

- Introductions (10 Minutes)
- Vehicle Networks: Vehicle Networks History (10 Minutes)
- Vehicle Networks: CAN Bus-History (10 Minutes)
- Vehicle Networks: CAN Bus-Physical Layer (30 Minutes)
- **Activity**: Create our Own CAN Bus (15 Minutes)
- **Activity**: Transmit Messages on CAN Bus (15 Minutes)
- --Break (20 Minutes) --
- Vehicle Networks: CAN Bus Frame (10 Minutes)
- **Activity**: Receive Messages on CAN Bus (30 Minutes)
- **Activity**: Reverse Engineer Controller Traffic (30 Minutes)
- **Activity**: DoS CAN Bus (15 Minutes)
- --Break (20 Minutes) --
- Diagnostics: Building a ISO 15765-2 Frame (30 Minutes)
- **Activity**: Fuzzing CAN Bus (30 Minutes)
- Security: Understanding Security Access (30 Minutes)
- **Activity**: Crack the Security, and WIN? (After Hours)

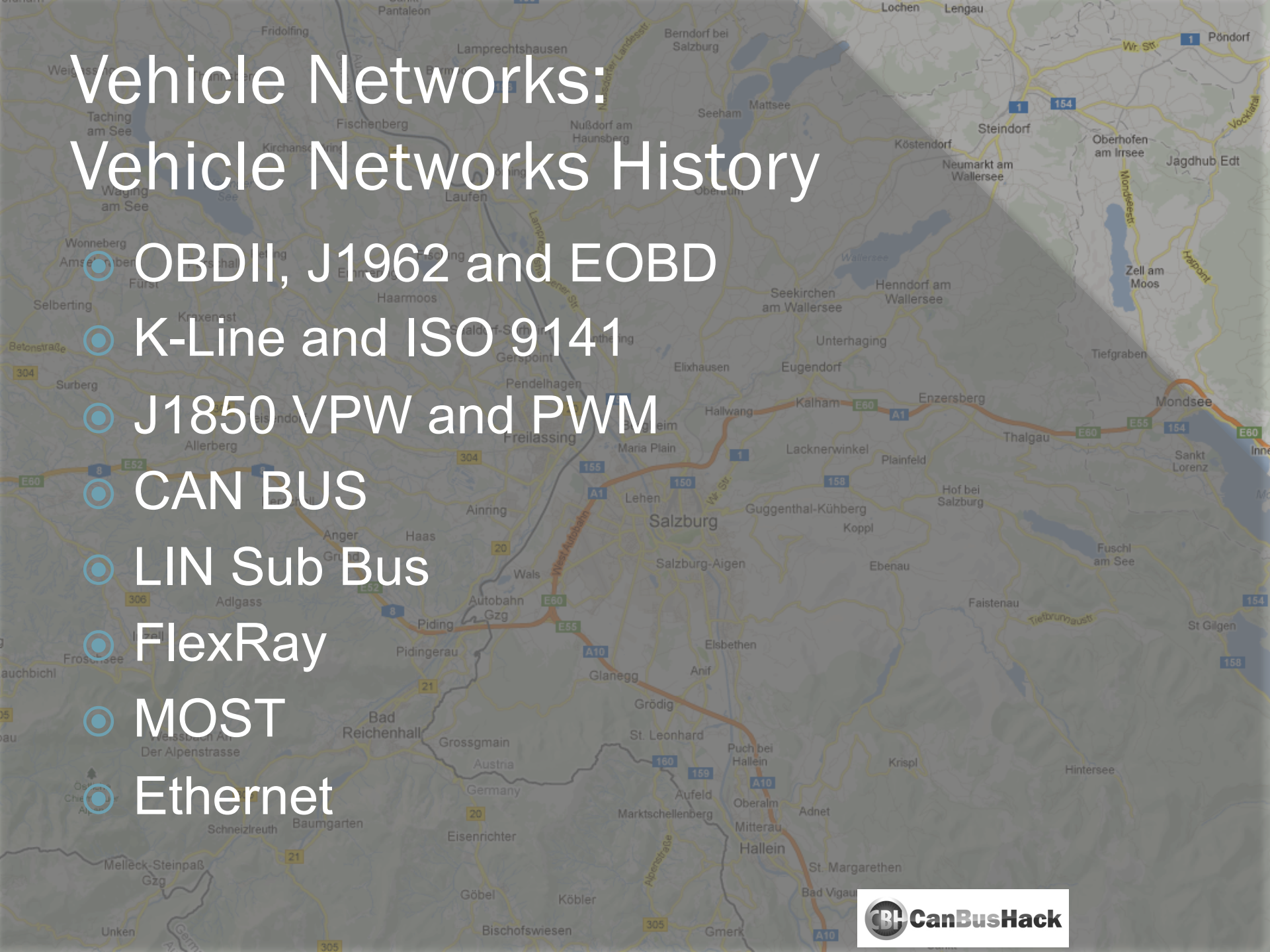
CanBusHack

Introductions

- Say hello if you'd like.
- Say no thanks if you'd like.

Vehicle Networks:
Vehicle Networks History

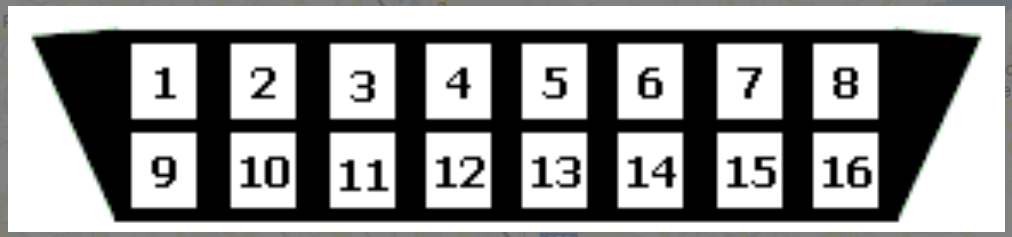
- OBDII, J1962 and EOBD
- K-Line and ISO 9141
- J1850 VPW and PWM
- CAN BUS
- LIN Sub Bus
- FlexRay
- MOST
- Ethernet



Vehicle Networks: OBDII, J1962, and EOBD

- OBDII is mandated diagnostic Method in USA for Vehicles release in 1996 and newer
- EOBD is mandated on 2001 and newer (petrol) and 2004 and newer (diesel).
- J1962 is the connector design, pinout, networks configuration, and connector location for the Diagnostic Connector in OBDII and EOBD vehicles.

Vehicle Networks: J1962 Connector Pinout



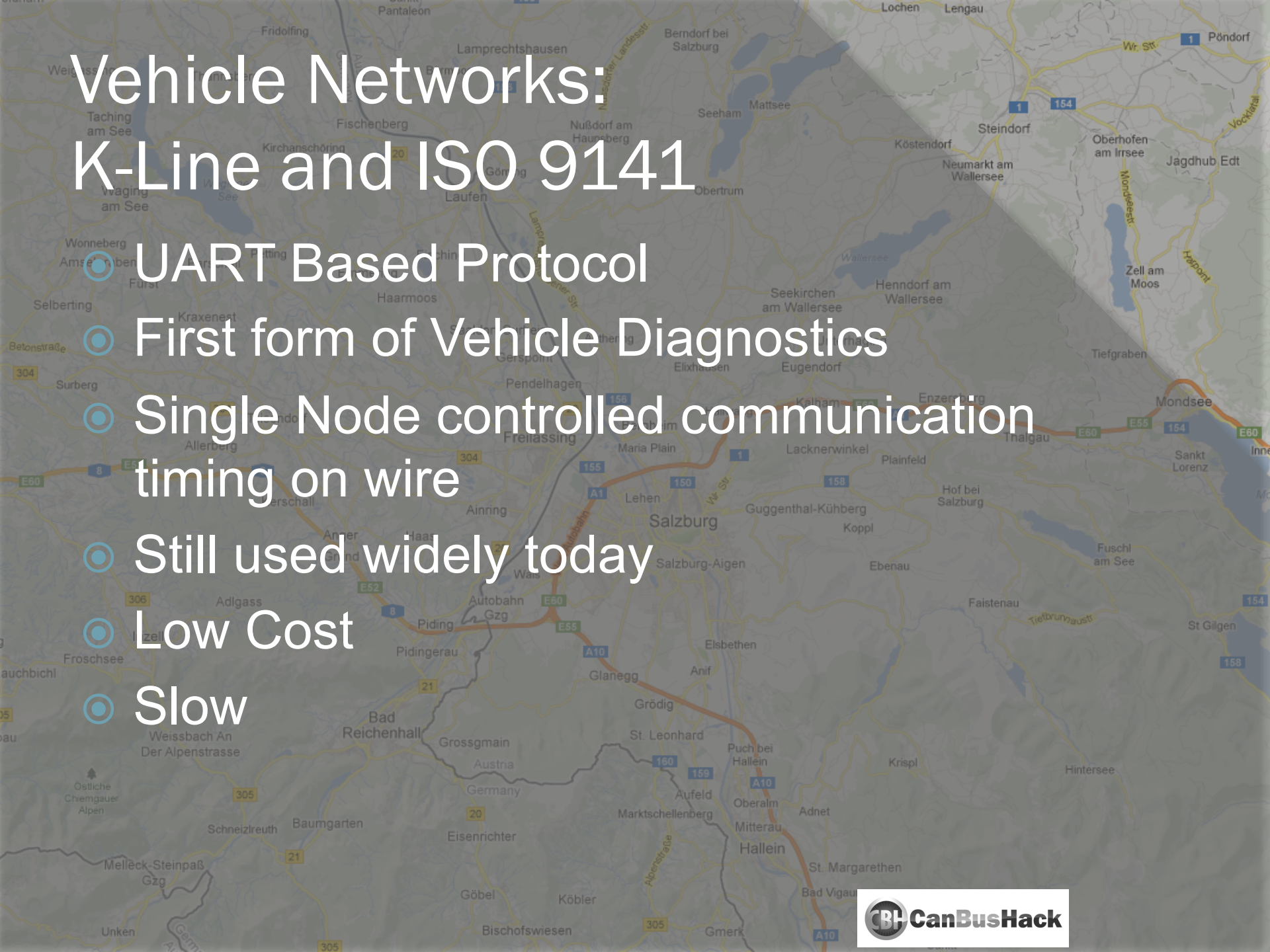
J1962 Pin	J1962 Pin Description
1	Discretionary* (GMLAN SW CAN Line)
2	+ line of SAE J1850
3	Discretionary* (GMLAN MS CAN H)
4	Chassis Ground
5	Signal Ground
6	Discretionary* (GMLAN HS CAN H)
7	K Line of ISO 9141-2
8	Discretionary*

J1962 Pin	J1962 Pin Description
9	Discretionary* (GM ALDL)
10	- line of SAE J1850
11	Discretionary* (GMLAN MS CAN L)
12	Discretionary*
13	Discretionary*
14	Discretionary* (GMLAN HS CAN L)
15	L line of ISO 9141-2
16	Un-switched Vehicle Battery Positive

CanBusHack

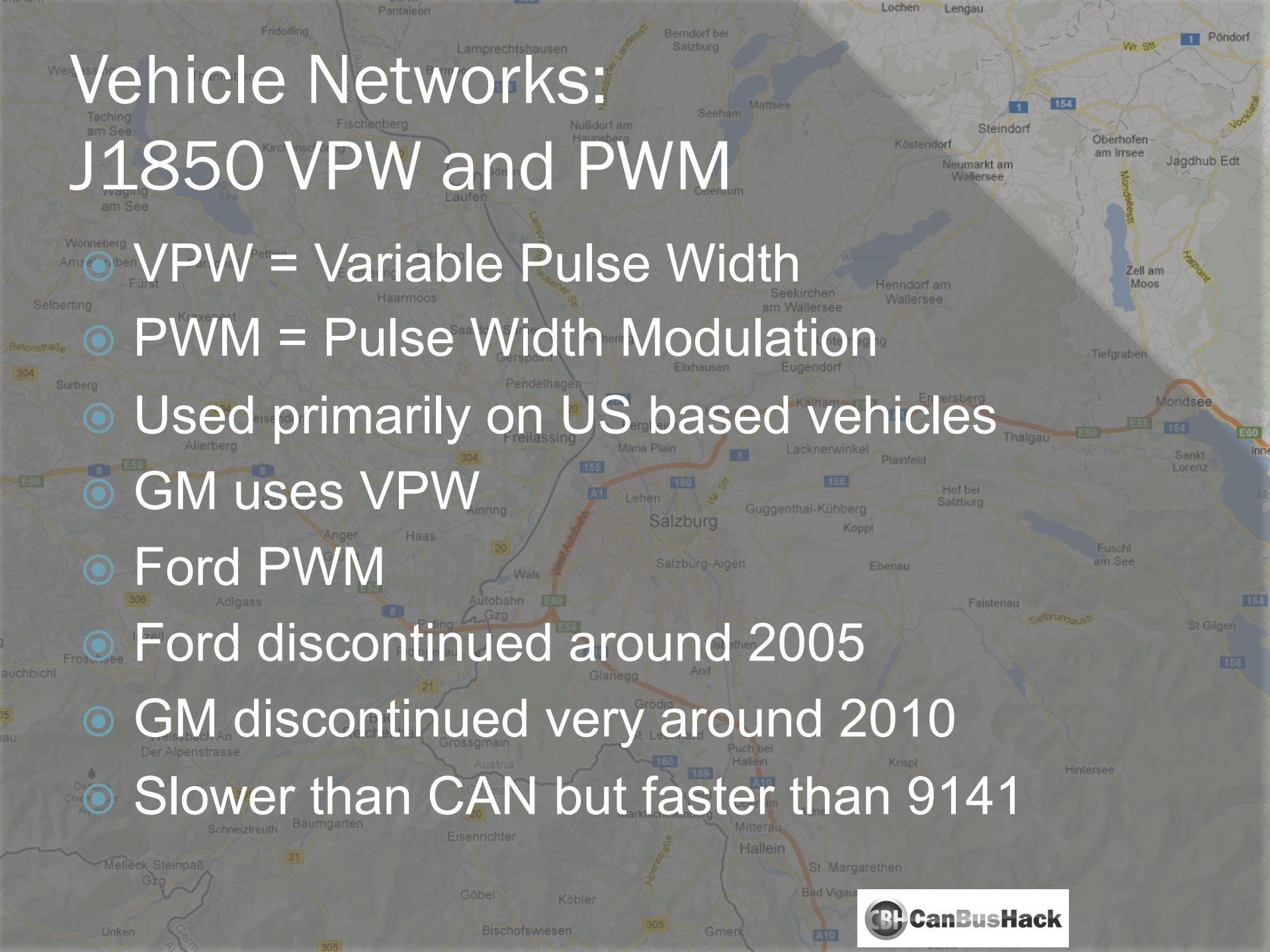
Vehicle Networks: K-Line and ISO 9141

- UART Based Protocol
- First form of Vehicle Diagnostics
- Single Node controlled communication timing on wire
- Still used widely today
- Low Cost
- Slow



Vehicle Networks: J1850 VPW and PWM

- VPW = Variable Pulse Width
- PWM = Pulse Width Modulation
- Used primarily on US based vehicles
- GM uses VPW
- Ford PWM
- Ford discontinued around 2005
- GM discontinued very around 2010
- Slower than CAN but faster than 9141



Vehicle Networks: LIN Sub Bus

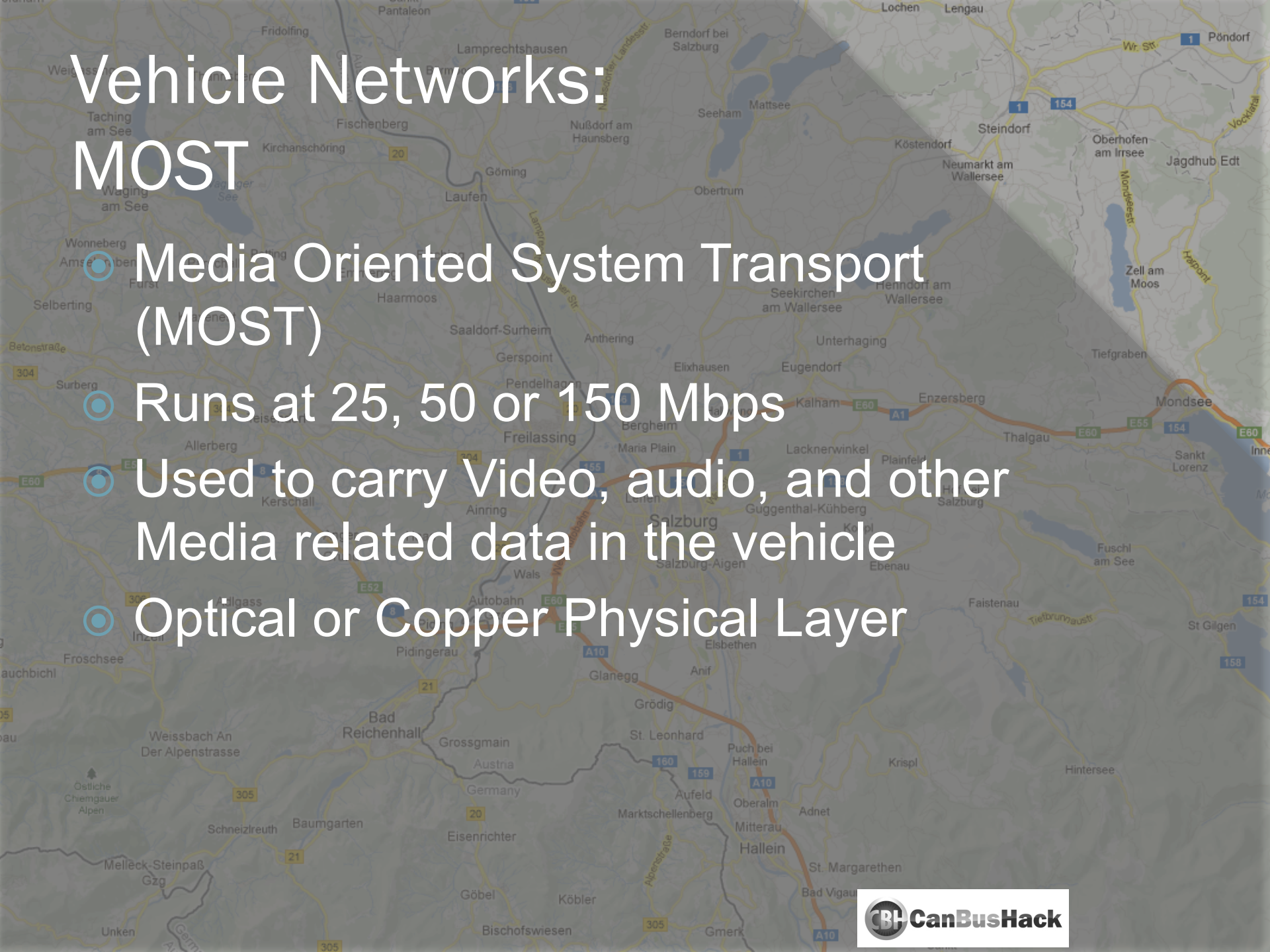
- Local INterconnect (LIN)
- Specification is feely downloadable from lin-subbus.org
- Latest Version is 2.2A
- Most OEMs use 2.1 or 2.0
- SAE Standard J2602/2 (Not Free)
- Used as a local network
- UART based with checksum

Vehicle Networks: FlexRay

- Created to answer the problems of CAN Bus
- Capable of redundancy
- Up to 10Mb (10x CAN Bus)
- Found on Audi, Bentley, BMW and Rolls-Royce
- Time Deterministic (Mostly)
- Costly to implement

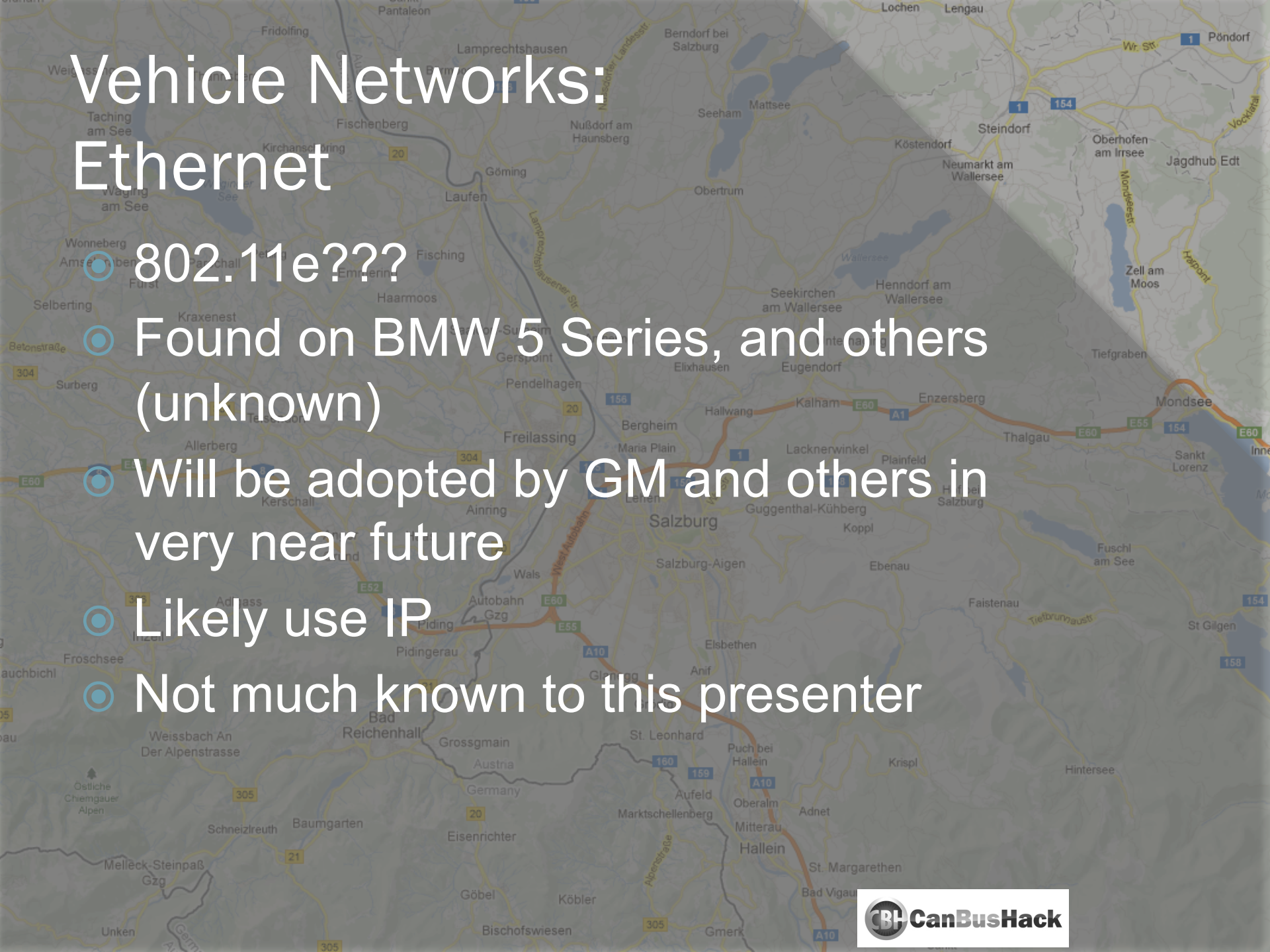
Vehicle Networks: MOST

- Media Oriented System Transport (MOST)
- Runs at 25, 50 or 150 Mbps
- Used to carry Video, audio, and other Media related data in the vehicle
- Optical or Copper Physical Layer



Vehicle Networks: Ethernet

- 802.11e???
- Found on BMW 5 Series, and others (unknown)
- Will be adopted by GM and others in very near future
- Likely use IP
- Not much known to this presenter



Vehicle Networks: CAN Bus-History

- Created by Bosch in 1983
- First Specification released in 1986
- First automobile uses CAN in 1992
- Generally adapted in early 2000's by the automotive industry
- Mandated OBDII network for MY2008 and newer in US

Vehicle Networks: CAN Bus-Physical Layer

- ISO 11898-2 defines a commonly used CAN Bus PHY
- ISO 11898-3 defines Fault Tolerant PHY
- J2411 is GM's Single Wire CAN (Also used on some older Hondas)

Vehicle Networks: CAN Bus-ISO 11898-2

- Most used physical layer
- Uses Unshielded Twisted Pair
- Allows for up to 64 Nodes
- Network Length inversely proportional to Baud Rate
- Up to 1Mbps; Most commonly at 500Kbps
- Typically rests at 2.5V
- Each wire has a name: CAN High and CAN Low
- Terminated at each end of Bus with 120 Ω

Activity:
Create our Own CAN Bus

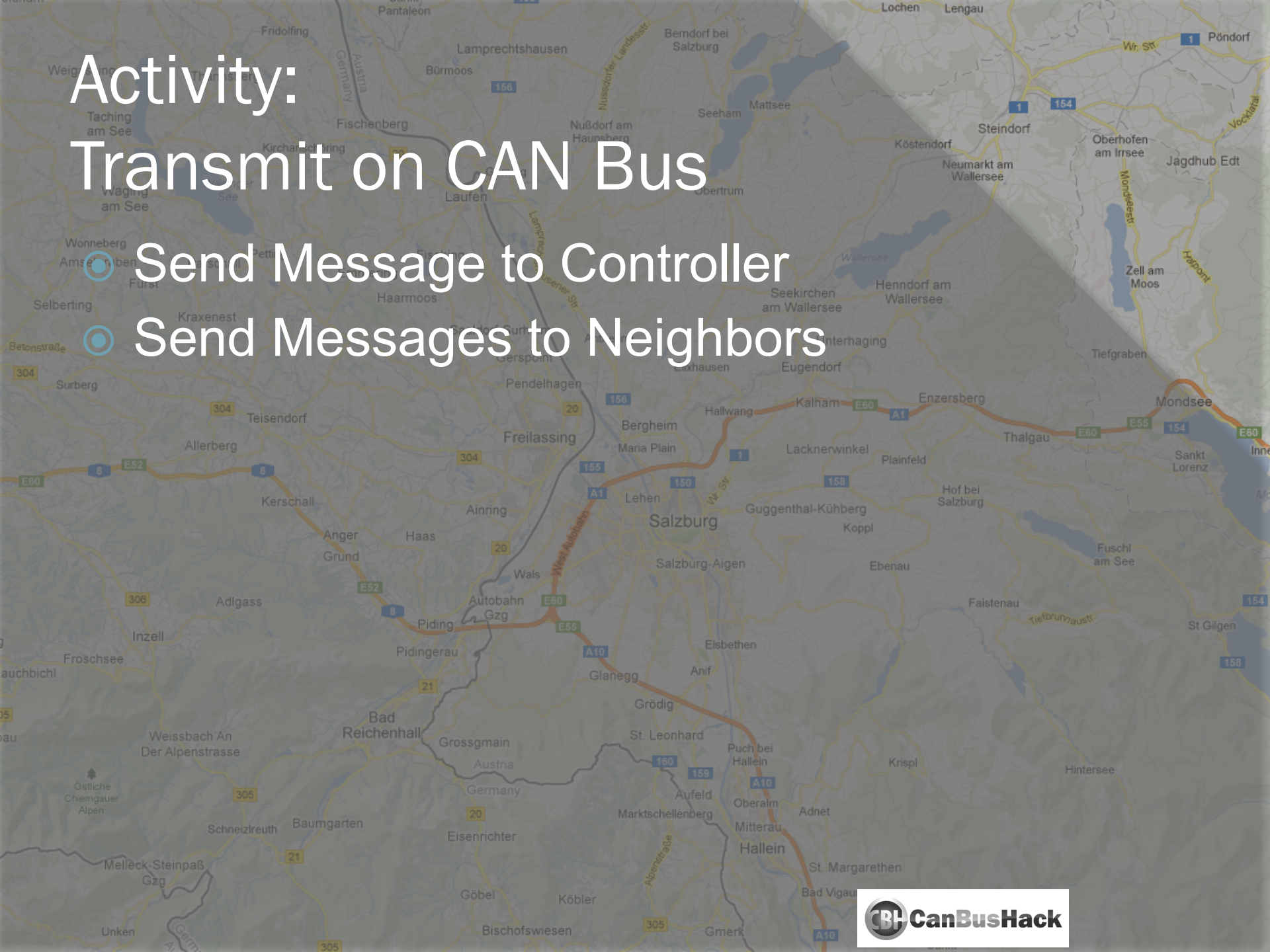
- Create an ISO 11898-2 Compliant network
- Connect USB to CAN devices
- Connect a hardware controller

Activity:
Create our Own CAN Bus

- Ensure Resistors are 120 Ohm
- Ensure hardware baud rates are set to 500Kbps
- CAN H is Yellow
- CAN L is Green
- Play!

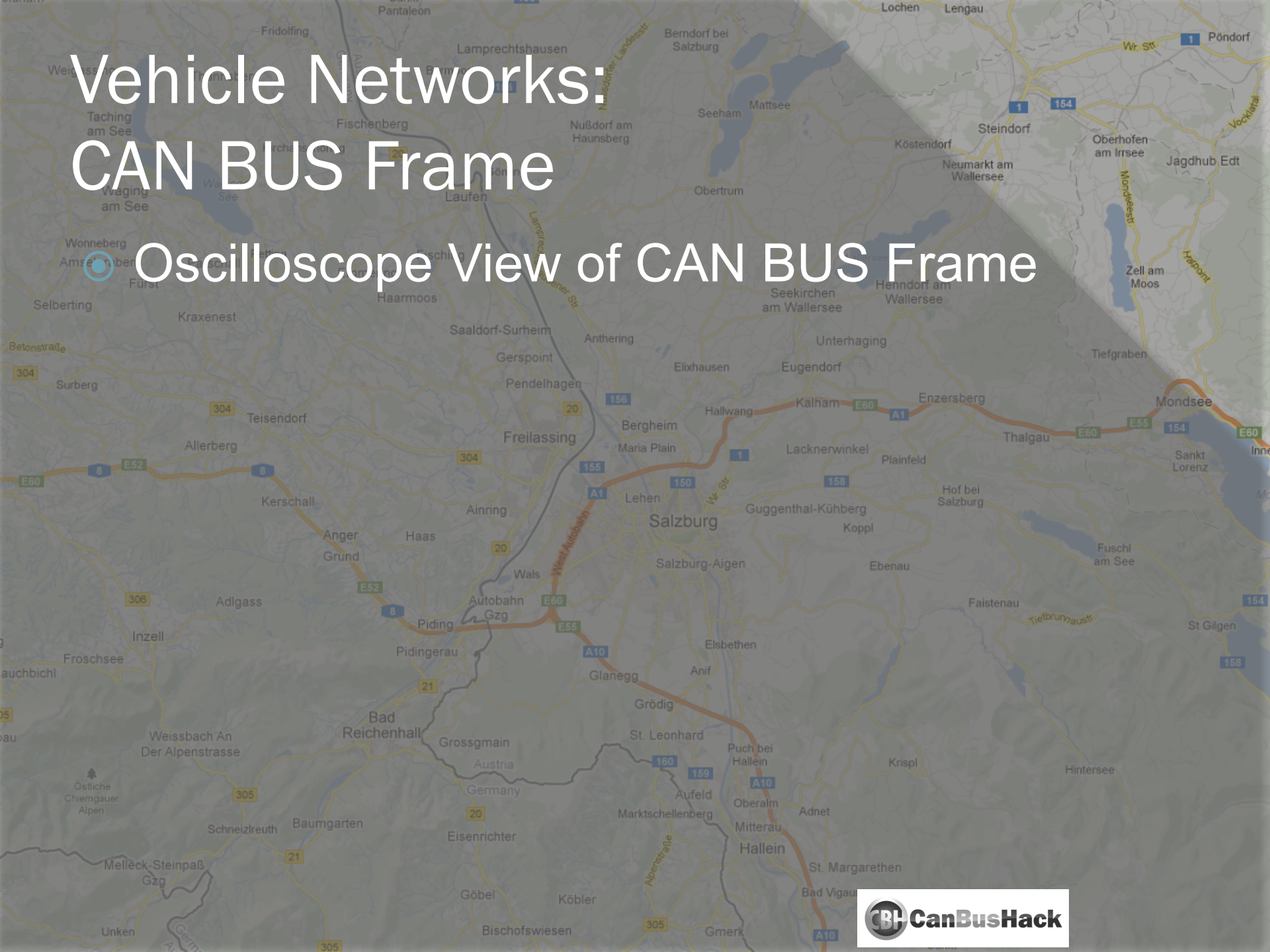
Activity: Transmit on CAN Bus

- Send Message to Controller
- Send Messages to Neighbors



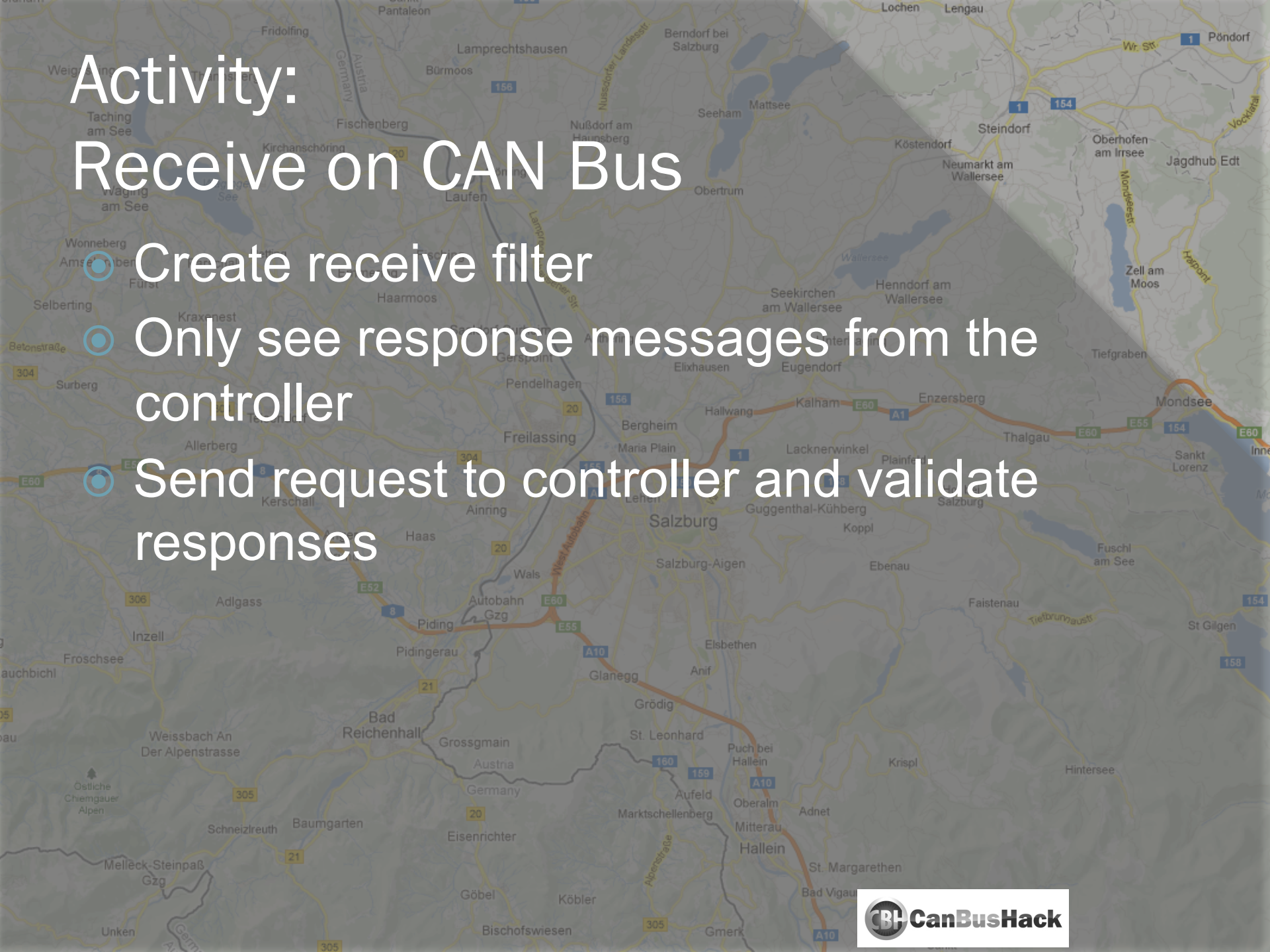
Vehicle Networks: CAN BUS Frame

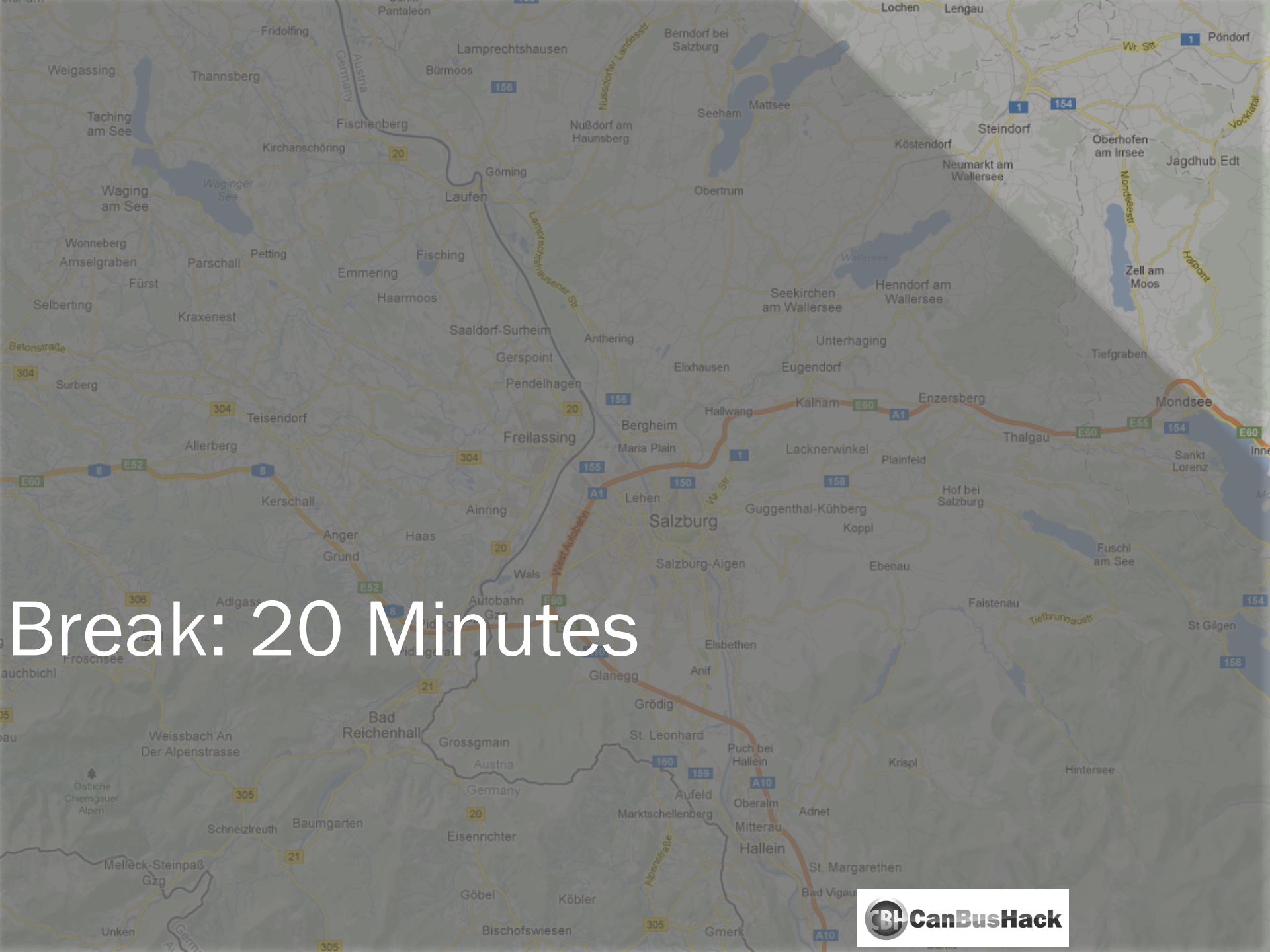
- Oscilloscope View of CAN BUS Frame



Activity: Receive on CAN Bus

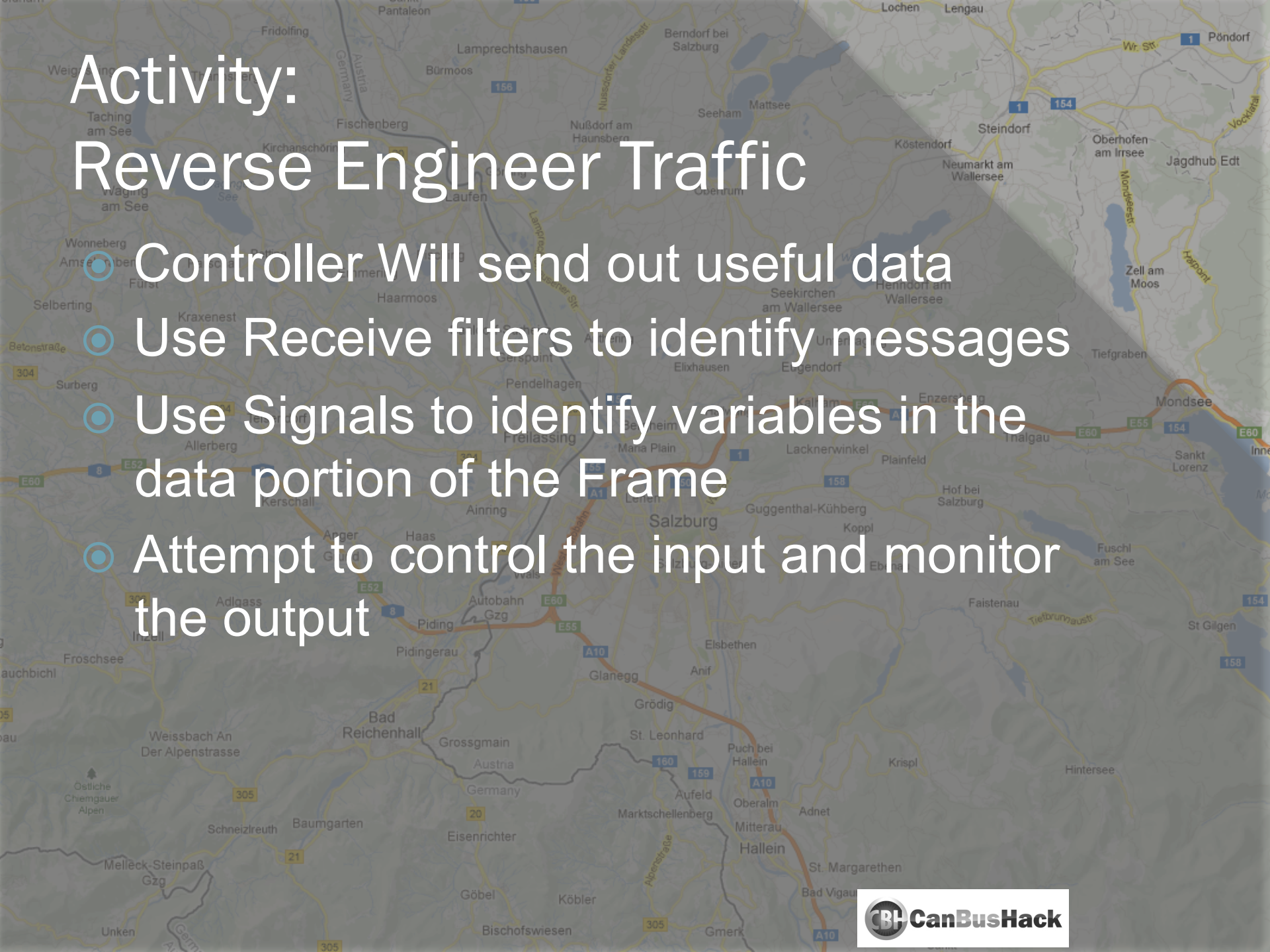
- Create receive filter
- Only see response messages from the controller
- Send request to controller and validate responses




Break: 20 Minutes
CanBusHack

Activity: Reverse Engineer Traffic

- Controller Will send out useful data
- Use Receive filters to identify messages
- Use Signals to identify variables in the data portion of the Frame
- Attempt to control the input and monitor the output



Activity:
DoS CAN Bus

- Send too many messages
- Send high priority message quickly
- Short the CAN Bus
  - CAN H to L
  - CAN H to Ground
  - CAN L to Ground
  - CAN H to Vbatt
  - CAN L to Vbatt

Diagnostics: Building a ISO 15765-2 Frame

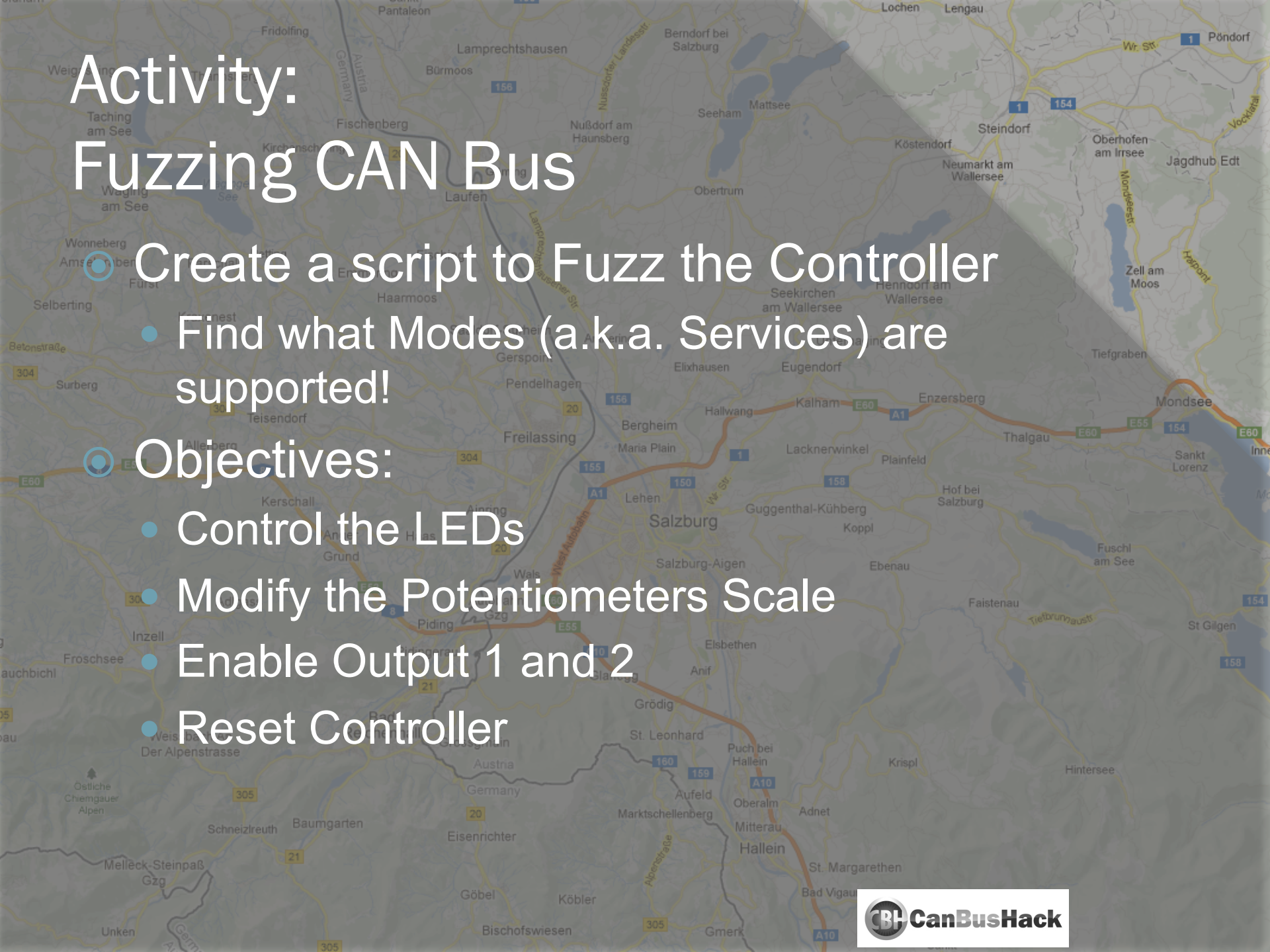
- Used as the Transport Layer on CAN BUS
- Can Send up to 4095 Bytes of Data.
- Uses Program Control Information (PCI) bytes for re-assembly
- First Data byte of message is PCI byte
  - If first nibble of byte = 0 then it is a single frame
  - If first nibble of byte = 1 then it is a multi-frame
  - If first nibble of byte = 2 then it is a consecutive frame
  - If first nibble of byte = 3 then it is a flow control frame

Building a ISO 15765-2 Frame: Single Frame

- Example:
  - 0x7E0 03 22 00 0C 00 00 00 00
- “03” will be a single frame with 3 bytes of interesting data

Activity: Fuzzing CAN Bus

- Create a script to Fuzz the Controller
  - Find what Modes (a.k.a. Services) are supported!
- Objectives:
  - Control the LEDs
  - Modify the Potentiometers Scale
  - Enable Output 1 and 2
  - Reset Controller



Security: Understanding Security Access

- There may be multiple security doors you will have to open
- Most things not behind security
- Module Re-programming is Always Behind Security
- Odometer Re-programming likely behind two levels of security

Security: Messaging

- Request Seed from Controller:
  - 0x7E0 02 27 01 00 00 00 00 00
- Controller Sends Seed
  - 0x7E8 05 67 01 8F 9D 3F 00 00
- Compute Key using Seed
  - Key' = h(Seed + StaticValue)
- Send Key
  - 0x7E0 05 27 02 33 7F 99 00 00
- Get Response from Controller
  - Positive 0x7E8 02 67 02 00 00 00 00 00
  - Negative 0x7E8 03 7F 27 22 00 00 00 00