SYSTEMS ENGINEERING
COLORADO STATE UNIVERSITY

Student CyberTruck Experience Manual

Jeremy Daily

May 23, 2020

A special thank you goes out to Urban Jonson at the National Motor Freight Traffic Association, Inc. (NMFTA) for being a champion of the program. His passion for cybersecurity and education was insiprational in creating the content and program called the Student CyberTruck Experience. The Executive Director the NMFTA, Paul Levine, deserves special thanks as he provided resources and guidance on bringing the Student CyberTruck Experience to life. Our initial sponsors at the University of Tulsa include the NMFTA, Geotab, and PeopleNet. The continued support from Geotab through Ryan Brander and Glenn Atkinson are gratefully appreciated.

The students who significantly contributed to this manual, both in writing and participating in the Student CyberTruck Experience include: Hayden Allen, Duy Van, John Maag, Ryan Corley.

Contents

Introduction to the Student CyberTruck Experience

In 2016, Urban Jonson of the National Motor Freight Traffic Association, Inc. () reached out to Dr. Jeremy Daily while he was teaching mechanical engineering at the Univeristy of Tulsa () to disuss some research findings related to heavy vehicle cybersecurity. In this conversation, Dr. Daily confirmed many of the hypotheses in the NNFTA whitepaper on the state of heavy vehicle cybersecurity. This converstation led to an invitation for Dr. Daily to attend the firs NMFTA Heavy Vehicle Cyber Security () meeting in Virginia. One of the results of the meeting was a recognition for the need to build the human talent needed to address the challenges assoicated with cybersecurity of heavy vehicles. This initiative is how the Student CyberTruck Experience came into being.

Learning Outcomes and Program Mission

To develop the talent necessary to improve the cybersecurity posture of the heavy vehicle.

Program Objectives

The Taxonomy of Educational Objectives (Bloom's Taxonomy)

Assessments and Exercises

Formative

Learning

Assessments are formative hands-on exercises where students

Summative

Research

Research is the top of the pyramid.

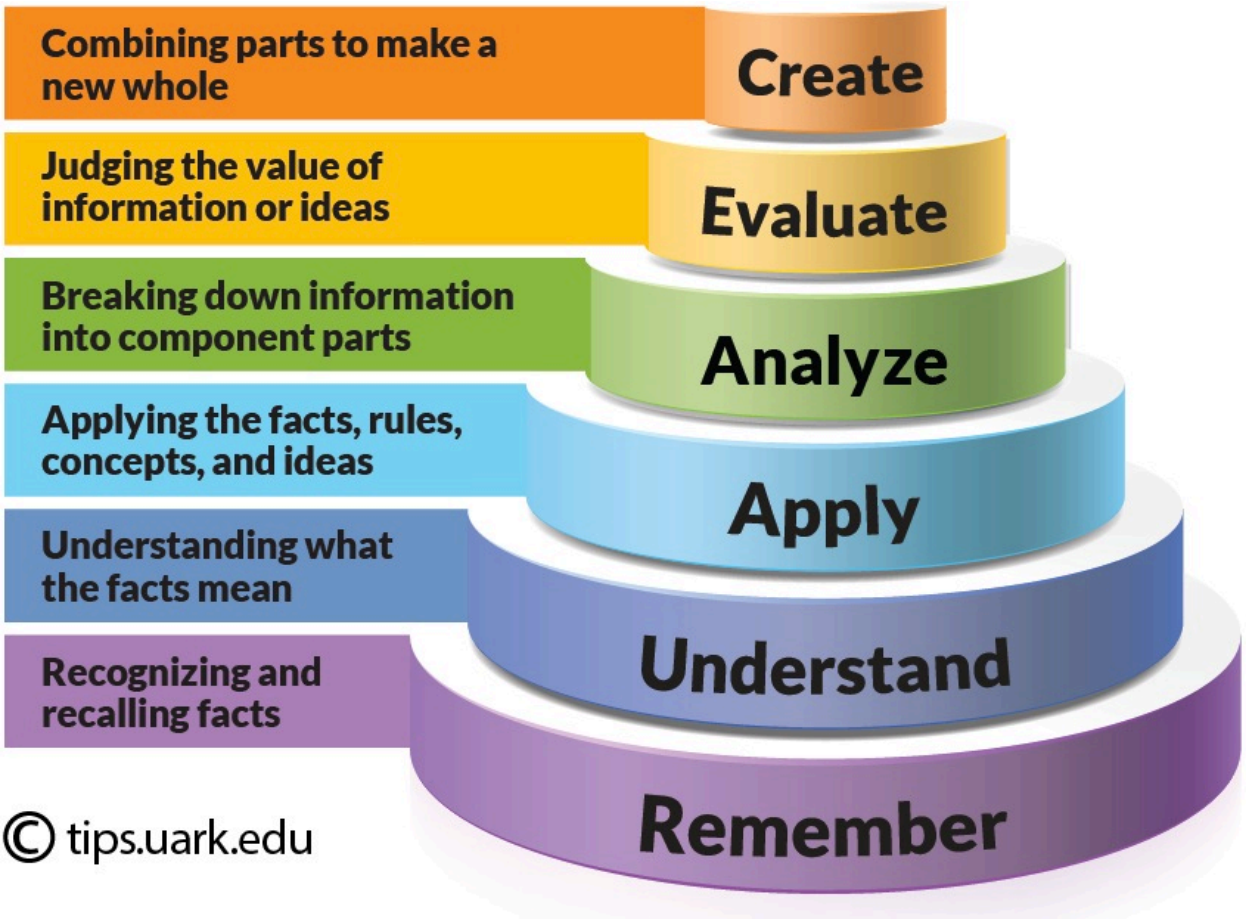

Figure 0.0.1: Bloom's Taxonomy from Jessica Shabatura.

1 Introduction to Trucking

1.1 Objectives

What is the trucking industry?

What is a truck?

Who makes trucks?

Cybersecurity for Trucking

1.2 Trucking Resources

The National Motor Freight Traffic Association, Inc. (NMFTA) has produced a great whitepaper [1]

2 Overview of Heavy Vehicle Systems

Weight Classes

3 Prototyping Electronics

The purpose of this chapter is to give students skills necessary to work with modern electronics and build their own hardware tools.

3.1 Intro to Microcontrollers

A **microcontroller (MCU)** is a tiny computer that can run one program at a time, over and over again. More often than not, they are connected to sensors and actuators allowing them to listen and interact with the physical world.

Sensors convert physical phenomena such as wheel speed, fuel level, and oil temperature into electrical signals. **Actuators** convert electrical energy back into physical energy such as linear motion, light, heat, and motor torque.

MCUs listen to sensors and talk to actuators. They decide what to do based on the instructions you have written and stored to its memory. MCUs and the devices connected to them form the basis of the **Electronic Control Unit (ECU)**.

3.2 Arduino and Teensy

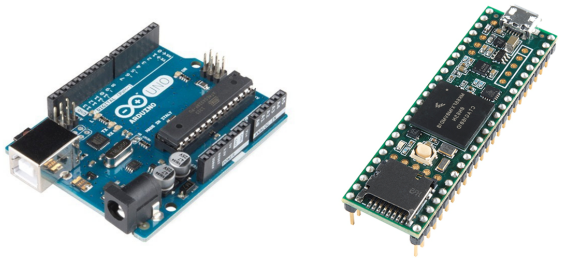

Figure 3.2.1: The Arduino Uno (left) and Teensy 3.6 (right)

Arduino offers some of the most recognizable open-source microcontroller boards today. This includes the Arduino Uno. When it comes to prototyping, these boards have many advantages including: widespread availability, low cost, thorough documentation, and a global community of users and forums. The Arduino

Uno uses the ATMEGA328P, has 32 KB of flash memory and runs at 16 MHz which are acceptable hardware specifications for simple projects.

Programs are written and compiled in the Arduino IDE program, a special text-editor, and uploaded via USB. When the Arduino is connected to an external power source it will run the instructions you uploaded.

The Teensy is another USB-based microcontroller development system. Although it is slightly more expensive than the Arduino, the 3.2 and 3.6 versions are preferred within CyTeX for their additional capabilities and features. Most notably, they support more communication protocols such as CAN, I2C, SPI, and Ethernet. The Teensy 3.6 footprint is roughly a third the size of the UNO, the MK66 Processor, has 1024 kB of flash memory, and runs at 180 MHz.

All programming is done via the USB port and the Teensy Loader Application. It is run automatically when using Verify or Upload within the Arduino software. Teensy Loader is available for Windows, Mac, and Linux.

Full documentation for the Teensyand Arduino UNOcan be found on their respective websites.

3.3 Intro to Microcomputers

A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. You *could* call it a "micro-computer." SBCs have the ability to run multiple programs, run an operating system, and can often support a simplified version of a typical desktop GUI.

The Raspberry Pi by the Raspberry Pi Foundation in the UK is by far the most common prototyping SCB. For automotive applications, the Beaglebone Black by Texas Instruments and the NXP S32K have been used within CyTeX. Both of these devices are capable of more advanced functionality.

3.4 Recommended Hardware Kit

Each student should have access to the parts in the basic kit. These items are minimum to successfully accomplish the programming and learning exercises.

Qty	Label	Description	Supplier	Supplier Part Number
1	A	Teensy 4.0 Development Board	PJRC	TEENSY40_PINS[1]
2	B	MPC2562 CAN Transceiver	Digi-Key	MCP2562FD-E/P-ND[2]
1	C	Solderless Breadboard	Digi-Key	BKGS-830-ND[3]
5	D	120 Ohm Axial Resistors	Digi-Key	S120CACT-ND[4]
1	E	Wiz850IO Ethernet Expansion Board	Digi-Key	1278-1043-ND[5]
1	F	30 Pack of Male-Male Breadboard Wires	Sparkfun	PRT-14284[6]
1	G	20 Pack of Male-Female Breadboard Wires	Sparkfun	PRT-12794[7]
1	H	Ethernet Cat6 Cable, 3 ft.	Sparkfun	CAB-08915[8]
1	J	USB micro Cable, 6 inch	Sparkfun	CAB-13244[9]
1	K	SOIC8 to DIP Converter Board	Sparkfun	BOB-13655[10]
1	L	ATECC608A Crypto Authentication Module	Digi-Key	ATECC608A-SSHDA-TCT-ND[11]
1	M	ATECC608 Crypto Co-Processor Breakout	Sparkfun	SPX-15838[12]
1	N	Trimpot 10K Ohm with Knob	Sparkfun	COM-09806[13]
1	P	Break Away Headers - Straight	Sparkfun	PRT-00116[14]
1	Q	Ambient Temperature Sensor	Sparkfun	SEN-14049[15]
1	R	Conductive 18 Compartment Organizer	Flambeau	C618[16]

Table 3.4.1: Basic Hardware Kit (Either use M or a the combination of K and L.)

Figure 3.4.1: Photograph Example Parts Kit

3.4.1 Basic Kit

Newer laptop computers may not have a physical Ethernet port, so a USB to Ethernet adapter may be necessary.

Exercise 3.1. [Knowledge] Download the Datasheet for the Teensy 4.0 processor. Answer the following questions:

1. What is the name of the processor?
2. What technology is the processor core built on?

[1] https://www.pjrc.com/store/teensy40_pins.html
[2] https://www.digikey.com/product-detail/en/microchip-technology/MCP2562FD-E-P/MCP2562FD-E-P-ND/4842807
[3] https://www.digikey.com/products/en?keywords=BKGS-830-ND
[4] https://www.digikey.com/product-detail/en/stackpole-electronics-inc/RNMF14FTC120R/S120CACT-ND/2617441
[5] https://www.digikey.com/products/en?keywords=Wiz%20850
[6] https://www.sparkfun.com/products/14284
[7] https://www.sparkfun.com/products/12795
[8] https://www.sparkfun.com/products/8915
[9] https://www.sparkfun.com/products/13244
[10] https://www.sparkfun.com/products/13655
[11] https://www.digikey.com/products/en?keywords=ATECC608A-SSHDA-TCT-ND
[12] https://www.sparkfun.com/products/15838
[13] https://www.sparkfun.com/products/9806
[14] https://www.sparkfun.com/products/116
[15] https://www.sparkfun.com/products/14049
[16] https://www.flambeaucases.com/18-compartment-box-1095.aspx

Qty	Description	Supplier	Supplier Part Number	URL
1	Teensy 4.1	PJRC	TEENSY41	[17]
1	Beagle Bone Black			
1	Quadrature Encoder with breakout board. Bournes PEC09			

Table 3.4.2: Advanced Hardware Kit

Qty	Description	Supplier	Supplier Part Number	URL
1	Smart Sensor Simulator 2	DG Technologies	TEENSY41	[18]
1	Beagle Bone Black			
1	Quadrature Encoder with breakout board. Bournes PEC09			

Table 3.4.3: Vehicle ECU Testing Kit

3. How many bits does the processor use when executing instructions?
4. How fast does the processor run?
5. How many CAN channels does the micoprocessor have?
6. What are the register number bases for the CAN Channels?

Exercise 3.2. [Comprehension] After reviewing the datasheet for the MCP2562, address the following questions:

1. Why do we need to connect the

3.4.2 Basic Tools

Multimeter Any modern digital multimeter is acceptable to get started. There are many on Amazon that are suitable.

Soldering Iron A

Tweezers

Wire Cutters

Wire Strippers

Carrying Tote

30 gauge Wire

3.4.3 Advanced Hardware Kits

3.5 Electronic Hardware Exercises

These exercises shouls serve to get you "up to speed" on Arduino hardware. Little to no computer programming experience is needed for this section, that comes later. However, you should know how to upload software in Arduino and view the console.

Exercise 3.3. Write a program to send CAN messages from one node to another.

Hints:

1. Check the continuity of each hookup wire before using it. They can be fragile and break easily.

3.5.1 Getting Started on Teensy Arduino (Stock Code that "should work")

3.5.2 Intro to CAN with Arduino (Move after CAN Section?)

3.5.3 Build a Breadboad Arduino (Move Challenge to Challenges Section)

Challenge: using the pieces below, send a CAN message.

Materials:

Breadboard Prototyping Wire ATmega328P MCP2551/MCP2562/MCP2558 MCP2515/MCP25625 LED 120ohm (2) or 60 ohm (1) resistor a 16MHz crystal 6 pin straight header a 10k resistor 18-22 pF capacitor (2)

Resources:

Use Datasheets to determine how to connect everything To find datasheets use google, digikey, and mouser Learning how to push code to the arduino: https://www.arduino.cc/en/Tutorial/ArduinoISP For help with SPI communication to MCP2515: http://avrbeginners.net/architecture/spi/spi.html

Challenge:

Send a CAN Message (Verify with oscilloscope) Decode the message from the oscilloscope to ensure that it is the mesage you told your arduino to send This can be done with Python Create an Altium Schematic of the layout that you created

[17] https://www.pjrc.com/store/teensy41.html
[18] https://www.pjrc.com/store/teensy41.html

4 Truck Systems for Computer Scientists

5 Computer Programming Fundamentals (Computer Programming for Engineers)

Computer programming may seem like a daunting field for beginner programmers. It does not need to be. At its core, computer programming is providing a set of instructions to a computer. Each line of code contains one or more instructions. Sometimes instructions interfere with each other and cause errors. Imagine telling someone who has never seen a sandwich before how to make a peanut butter and jelly sandwich. The steps may be:

1. Get Bread, Peanut Butter, Jelly, Knife, and Plate.
2. Place plate.
3. Place two pieces of bread side by side on plate. Call them Slice A and Slice B.
4. Put peanut butter on knife.
5. Use knife to spread peanut butter on Slice A.
6. Clean knife.
7. Put jelly on knife.
8. Use knife to spread jelly on Slice B.
9. Combine Slice A with Slice B by placing Slice B on Slice A.
10. Eat Sandwich.
11. Clean Knife
12. Clean Plate

Provided no external changes, this set of instructions will always lead to a proper peanut butter and jelly sandwich. However, some people may argue about if you should use grape jelly or strawberry jelly, chunky or smooth peanut butter. Some people are adamant that you must put Jelly on Slice A, never peanut butter. Some argue that if one starts with a loaf of bread instead of slices then it produces a better sandwich. These options represent the different ways that people program. There are many different sets of instructions that produce peanut butter and jelly sandwiches. Programming is very smiliar. There are many different ways to code a program that will lead to the same result.

However, there are ways which are more efficient than others, and there are ways to make code easier to understand. When writing code, it is important to write with a consistent programming style that allows others to understand your code. Code must be maintained over time, and may transition between programmers.

This chapter should serve to introduce the basics of computer programming as a platform for the Cyber Truck Experience program.

5.1 Arduino Programming Language

5.2 Python 3 Programming Language

An important tool in any hackers toolkit is the ability to interact with collected data. In this section, we will be using a programming language called Python to help with this.

5.2.1 Background

Python is a general purpose, versatile, and popular programming language. It is great as a first language because it is concise and easy to read. It is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and data science applications.

This is different than what we have done to this point in Arduino. We will generally use the Arduino devices to directly interface with systems. These devices will then send data back to a computer or other device running a python script.

5.2.2 Codecademy

We will begin by having you learn the basics of python. We will be using an external system to aid in this. Navigate over to hereand create an account using your email and fill out any information they require. Please ensure that you do not pay for this service at this time. We will only need the services that the free version offers.

Please complete this course. Take notes throughout the course over functions that may be useful (importing and exporting files, searching through lists, data structures, etc.) Periodically, there will be additional trial material to supplement your learning.

5.2.3 Editors and IDE

5.2.4 Threading

5.2.5 PySerial

5.2.6 matplotlib

5.2.7 Making GUIs with PyQt5

5.3 C++ Programming Language

5.3.1 Codecademy

5.4 Computer Programming Fundamentals Exercises

6 Serial Communication with SAE J1708 and J1587

Exercise 6.1. CAN Frame Decoding

Capture a CAN Frame using an oscilloscpope on a J1939 network and decode it according to SAE J1939.

Exercise 6.2. Read Live Engine RPM Challenge

Using a BeagleBone Black and a Truck Cape, connect to an engine contoller that is broadcasting non-zero engine RPM. Gather this data using candump. Interpret the raw CAN frames and extract information for Engine RPM, or J1939 SPN 190. Plot 20 seconds of changing RPM with matplotlib. Print the properly labeled plot to PDF and show it to your instructor. Objectives

Learn how to interface with Linux SocketCAN and can-utils Be able to look up a signal definition in the J1939 Digital Annex (spreadsheet) Use grep to search for specific strings from a candump Have a reliable CAN datalogger for use in future projects Plot data using matplotlib in Python.

Suggested Materials

This exercise can be run with any Linux device with CAN hardware. An example of a commercial product with these features is the DG Technologies' Beacon device. An example of a hand built project is the BeagleBone Black with a TU TruckCape. Resources

J1939DA Internet Access (you may want to share your PC's connection sharing)

Exercise 6.3. Man in the Middle

Build a man-in-the middle board and box that takes CAN signals into one can channel and sends them out on another. Start a diagnostics session with a PC and RP1210 device to perform maintenance. Create a forwarding system that inspects and forwards network traffic in both directions. Attempt to hijack a a diagnostic session and affect a parameter change started with the PC diagnostics software.

Using DDEC Reports, try to prevent resetting the CPC clock during a data extraction on a CPC.

7 CAN Communication with SAE J1939

Exercise 7.1. CAN Frame Decoding

Capture a CAN Frame using an oscilloscpope on a J1939 network and decode it according to SAE J1939.

Exercise 7.2. Read Live Engine RPM Challenge

Using a BeagleBone Black and a Truck Cape, connect to an engine contoller that is broadcasting non-zero engine RPM. Gather this data using candump. Interpret the raw CAN frames and extract information for Engine RPM, or J1939 SPN 190. Plot 20 seconds of changing RPM with matplotlib. Print the properly labeled plot to PDF and show it to your instructor. Objectives

Learn how to interface with Linux SocketCAN and can-utils Be able to look up a signal definition in the J1939 Digital Annex (spreadsheet) Use grep to search for specific strings from a candump Have a reliable CAN datalogger for use in future projects Plot data using matplotlib in Python.

Suggested Materials

This exercise can be run with any Linux device with CAN hardware. An example of a commercial product with these features is the DG Technologies' Beacon device. An example of a hand built project is the BeagleBone Black with a TU TruckCape. Resources

J1939DA Internet Access (you may want to share your PC's connection sharing)

Exercise 7.3. Man in the Middle

Build a man-in-the middle board and box that takes CAN signals into one can channel and sends them out on another. Start a diagnostics session with a PC and RP1210 device to perform maintenance. Create a forwarding system that inspects and forwards network traffic in both directions. Attempt to hijack a a diagnostic session and affect a parameter change started with the PC diagnostics software.

Using DDEC Reports, try to prevent resetting the CPC clock during a data extraction on a CPC.

8 Princples of Cybersecurity

The Triads of Cybersecurity

Confidentiality, Integrity, and Availability

Or,

Authentication, Authorization, and Auditing

8.1 Introduction to Cryptography

8.2 Message Authentication

8.3 Encryption on CAN

9 Heavy Vehicle Digital Forensics

Nomenclature

HVCS Heavy Vehicle Cyber Security

NMFTA National Motor Freight Traffic Association, Inc.

TU The University of Tulsa

Bibliography

[1] “A survey of heavy vehicle cyber security,” tech. rep., National Motor Freight Traffic Association, Inc., 1001 North Fairfax, Suite 600, Alexandria, Virginia 22314, September 2015. 8