Hardware Reverse Engineering

2024 Student Pre-Class Exercise

May 15, 2024

Instructor: Bill Hass
<billhass@umich.edu>

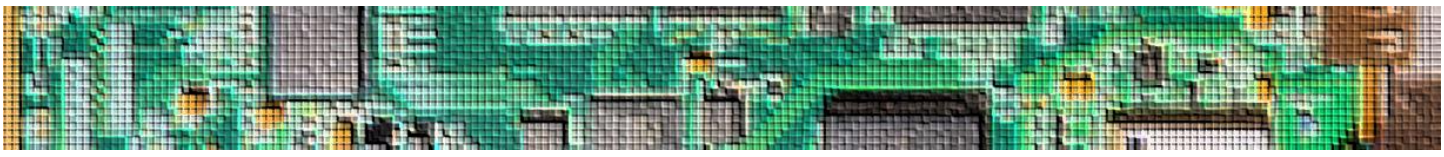

Table of Contents

1 Pre-Class Exercise (~60 minutes)

1.1 Introduction

This hardware reverse engineering (HW:RE) class is an introductory-level course intended not as a stepping stone, but as a diving board into the concepts and techniques used for HW:RE. By using real-world electronic control units (ECUs) with approachable and affordable open-source tooling, you will be equipped to confidently apply the learned concepts directly to other embedded systems you encounter.

1.2 Goals

This pre-class exercise will focus on the initial "Discovery" step of HW:RE. The high-level goal of the pre-class exercise is to learn and practice component identification by creating a bill-of-materials (BOM). Specifically:

1. Identify non-volatile memory component(s)
2. Determine the location and pin-out for UART transceiver TXA/RXA
3. Determine the location and pin-out for microcontroller JTAG TDI, TDO, TCK

By achieving these goals, you will lay the foundation necessary to succeed during the in-person portion of the class.

During the in-person portion of the class, you will get hands on with the same ECU covered during this exercise. You will gain familiarity with essential electronics tools, probe the device to identify circuits, practice soldering, discover how to power the ECU and how to get data off of it.

1.3 Background

Hardware reverse engineering (HW RE) is an iterative process to build-up hardware domain knowledge about an unfamiliar system. We iterate over these three steps: (1) Discovery; (2) Planning; & (3) Experimentation.

(1) Discovery - ***information gathering and documentation***
- Reference searches are carried out (perhaps somebody has already reverse engineered your target device or something similar!)
- Physical components and markings are identified
- A component list, aka bill-of-materials (BOM) is created

Through discovery, the HW RE engineer knows about connectors, external memory chips, microcontrollers, external and internal network interfaces, and miscellaneous/benign components on the target.

(2) Planning - considers information from the discovery phase to ***chart a path towards a goal, consider new goals, and prioritize next steps***
- From the discovery phase, you may have discovered hidden components or connectors that will make your job easier, or you may have noticed security features that dissuade you towards other low hanging fruit.

At the end of this phase, the HW RE engineer has a prioritized list of interesting things to try (a plan) and has a better feeling of what might work and what won't.

(3) Experimentation - ***executes parts of the plan to achieve a goal, aids in discovering more information, and/or verifies assumptions from the prior phases***.

- This phase may involve powering up the unit for the first time and taking measurements.

After experimentation, the HW RE engineer might know what the different power domains are, what external and internal communication interfaces are active, and/or that a particular circuit or pinout is what they thought it was.

1.4 Getting Started

First contact with a new component often involves carefully observing how it is constructed by visual inspection, poking, prying, and prodding. Like peeling back a metal onion, you may need to unscrew fasteners, peel apart adhesives, and pry apart clips to get to the electronics within. During class, you will have the hardware in front of you, but for this exercise you will use the included high-resolution image of the victim device's printed circuit board (PCB). Go ahead and open the image now: "victim-device-high-res-top-cthwre-2024.png". ***Note**: The image is a composite to ensure better quality focus on all areas of the board. You may notice artifacts, especially in the center of the image, but they will not prevent you from completing this exercise.*

As you gain access to the electronics, take some time to look carefully at them to see if you recognize anything or see any patterns. If you look closely, you will see a white printing of letters and numbers on the green surface of the PCB called a silkscreen. Like comments in software code, hardware engineers can use silkscreen to comment on the PCB. Commonly, silkscreen is used during development and debugging for adding reference designators so engineers can easily reference a board's components (e.g. R30 needs to be 125kΩ instead of 250kΩ). An "R" might stand for "resistor," "C" for "capacitor," "L" for "inductor," "J" for "jack," "TP" for "test point," and so-on. IEEE 315 standard has a list of these reference designators. Even without silkscreen, you can learn a lot about a system just by looking at it.

1.5 Chip Identification

Now you will build a "Bill-of-Materials" (BOM). An example is shown in Figure 1. This is one of the HW:RE's first major contributions to the overal security assessment.

Silkscreen/ID	IC Marking	Vendor	Component Type	Datasheet
U601	TIM-5H-0-004	Ublox	GPS Transceiver	https://u-blox.com/c
U805	POWR1014	Lattice	Power Supply Supervisor	lattice_latt-s-a00013

Figure 1: Example BOM sufficient for HW:RE

1.5.1 Step-by-Step Advice

1. Bigger chips on the circuit board are usually more complex and interesting things like processors and memory, so it can be useful to start with those.
2. Write down the silkscreen reference or ID and the IC marking.
3. Enter the IC marking directly into a search engine. Useful resources: https://duckduckgo.com/; https://octopart.com; http://www.smdmark.com/en-US/; https://www.elnec.com/en/support/ic-logos/.
4. Try to find the datasheet on the internet and use it to fill in the other columns.
5. Repeat until you identify every chip on the board. For this exercise it's sufficient to find the flash, UART transceiver, and the MCU.

1.6 Parsing Datasheets

Datasheets contain a plethora of technical information, but not all of it is relevant for HW:RE right away. Keyword searches are crucial for navigating a datasheet. Use the datasheets to accomplish the goals of this exercise listed in Section 1.2.

1.6.1 Step-by-Step Advice

1. Search for a block diagram which will shed some light on what is within the IC and how it may be connected to other components on the PCB (e.g. via I2C, SPI, UART, or USB).
2. Do a keyword search to see if anything interesting turns up using: debug, program, jtag, and security.
3. Look for pin descriptions and how they map to the physical package. For instance, this will tell you where a UART RX/TX or JTAG TDI, TDO, TCK interface is located physically on the device.
4. Look at recommended operating conditions so you can know how to safely power the device later on.

1.7 Questions

What is the BOM entry (Silkscreen/ID, IC Marking, Vendor, Component Type, Datasheet) for the non-volatile memory component(s):

ANSWER:

Silkscreen/ID	IC Marking	Vendor	Component Type	Datasheet
U903	S29GL256P11TFI02	Spansion / Infineon	Flash Memory	Datasheet Link

What is the BOM entry (Silkscreen/ID, IC Marking, Vendor, Component Type, Datasheet) for the UART transceiver?

ANSWER:

Silkscreen/ID	IC Marking	Vendor	Component Type	Datasheet
U501	ST16C554DJ3	Exar / MaxLinear	UART TRX	Datasheet Link

What pins (e.g. pin 1 & pin 3) are mapped to the UART transceiver's TXA/RXA?

ANSWER:

TXA = pin 17 RXA = pin 7

What is the BOM entry (Silkscreen/ID, IC Marking, Vendor, Component Type, Datasheet) for the microcontroller?

ANSWER:

Silkscreen/ID	IC Marking	Vendor	Component Type	Datasheet
U401	LH7A400	NXP	ARM Micro	Datasheet Link

What pins (e.g. pin 1, pin 2, & pin 3) are mapped to the microcontroller's JTAG TDI, TDO, TCK?

ANSWER:

TDI = pin A1 TDO = pin B2 TCK = pin B1