Akhila Parankusham

Tagliatela College of Engineering, University of New Haven

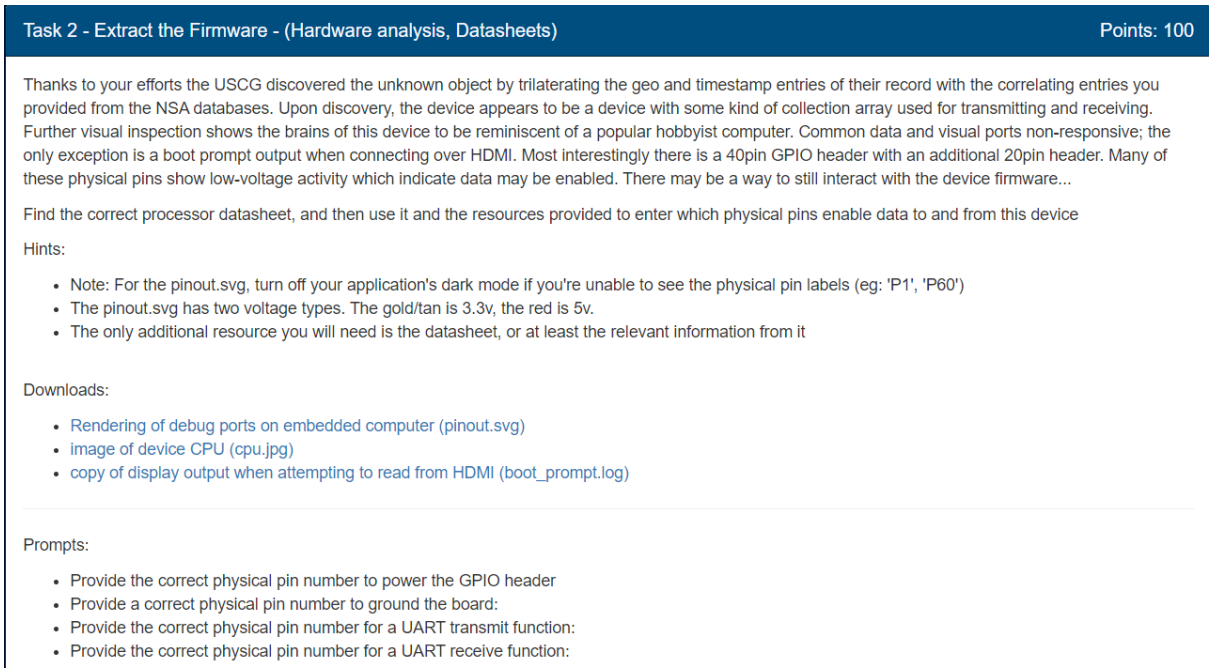
CSCI-4448-01: Reverse Engineering.

Professor: Robert Schmicker

NSA CODEBREAKER CHALLENGE -2023

Overview:
The NSA Codebreaker Challenge 2023 is an annual cybersecurity competition organized by the National Security Agency. It involves participants tackling various cybersecurity challenges, including cryptography, reverse engineering, and network analysis, aiming to solve real-world problems. Participants compete individually or in teams, showcasing their technical skills while learning from industry-standard scenarios.

Task 2:



Task 2 - Extract the Firmware - (Hardware analysis, Datasheets) Points: 100

Thanks to your efforts the USCG discovered the unknown object by trilaterating the geo and timestamp entries of their record with the correlating entries you provided from the NSA databases. Upon discovery, the device appears to be a device with some kind of collection array used for transmitting and receiving. Further visual inspection shows the brains of this device to be reminiscent of a popular hobbyist computer. Common data and visual ports non-responsive; the only exception is a boot prompt output when connecting over HDMI. Most interestingly there is a 40pin GPIO header with an additional 20pin header. Many of these physical pins show low-voltage activity which indicate data may be enabled. There may be a way to still interact with the device firmware...

Find the correct processor datasheet, and then use it and the resources provided to enter which physical pins enable data to and from this device

Hints:

- Note: For the pinout.svg, turn off your application's dark mode if you're unable to see the physical pin labels (eg: 'P1', 'P60')
- The pinout.svg has two voltage types. The gold/tan is 3.3v, the red is 5v.
- The only additional resource you will need is the datasheet, or at least the relevant information from it

Downloads:

- Rendering of debug ports on embedded computer (pinout.svg)
- image of device CPU (cpu.jpg)
- copy of display output when attempting to read from HDMI (boot_prompt.log)

Prompts:

- Provide the correct physical pin number to power the GPIO header
- Provide a correct physical pin number to ground the board:
- Provide the correct physical pin number for a UART transmit function:
- Provide the correct physical pin number for a UART receive function:

Executive Summary:

- **Discovery and Initial Description:** Using trilateration of location and temporal data from their records and comparing it with information from the NSA databases, the USCG found an unidentified object. The object that was found appears to be a transmitting and receiving device with a collection array of some kind.
- **Visual Inspection:** Upon closer examination, the device's interior parts bear similarities to those of a well-known hobby computer. Except for an output via HDMI, the majority of the device's widely used data and visual ports are non-responsive.
- **Physical Attributes and Components:** There is a 40-pin General Purpose Input/output (GPIO) header on the device, in addition to a 20-pin header. A few physical pins on the headers appear to be seeing low voltage activity, which suggests that data transfer may be occurring.
- **Opportunity for Further Interaction:** It has been suggested that communication with the device firmware may still be feasible even in the case of non-responsive ports.

- **Task Requirements:** To determine the precise physical pins in charge of data transfer, you must locate the relevant datasheet for the device's processor and use it in conjunction with the resources supplied.

Provided Resources:

- **pinout.svg:** This file shows a rendering of the debug ports on the embedded computer, denoting physical pin labels and two different voltage types (3.3v and 5v).
- **cpu.jpg:** An image of the device's CPU.
- **boot_prompt.log:** A copy of the display output obtained when attempting to read from HDMI.

Task Prompts:

- Identify the correct physical PIN number to power the GPIO header.
- Determine the correct physical pin number to ground the board.
- Find the correct physical pin number for a UART (Universal Asynchronous Receiver-Transmitter) transmit function.
- Identify the correct physical pin number for a UART receive function.

Overall, the objective is to leverage the provided CPU image, pinout.svg, and potentially the datasheet of the device's processor to accurately identify and label the physical pins responsible for powering, grounding, and enabling UART transmit and receive functions on the device.

Files:

- pinout.svg
- cpu.jpg
- boot_prompt.log

Task Steps:

Step 1: cpu.jpg

The CPU label, BCM2837RIFBG, corresponds to a Broadcom BCM2837 system-on-a-chip (SoC) found as the primary processor within both the Raspberry Pi 3 Model B and Raspberry Pi 3 Model B+. Online research under "BCM2837RIFBG datasheet" confirms this identification. Noted the physical characteristics and possible pin locations based on the image.

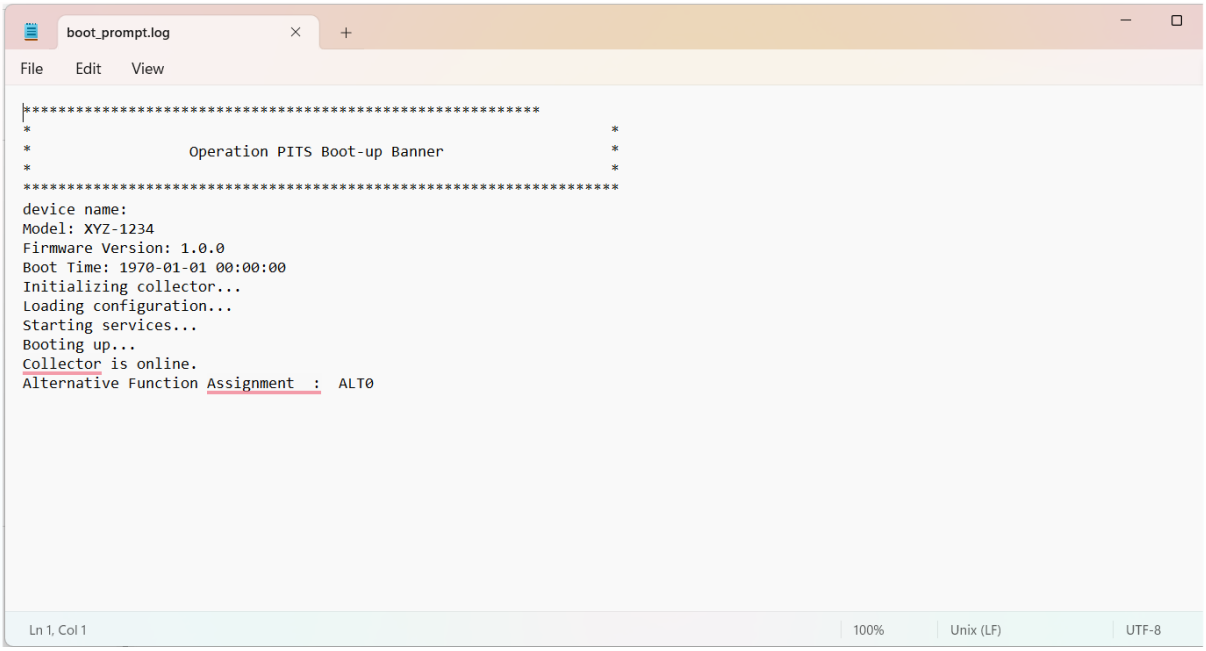
The summary of the BCM2837RIFBG:

- Quad-core 64-bit ARM Cortex-A53 CPU clocked at 1.2GHz
- VideoCore IV GPU @ 400MHz
- 1GB LPDDR2 RAM
- 802.11n wireless LAN and Bluetooth 4.1 BLE
- 4 USB 2.0 ports
- 40-pin GPIO header
- Camera and display interfaces
- MicroSD card slot
- HDMI port
- Composite video port

Step 2: Search for Datasheet:

- Searched online using the CPU's markings "BCM2837RIFBG" to find the datasheet.
- Located and accessed the datasheet from a reliable source, such as the manufacturer's website or technical documentation repositories.
- Interpreted the technical specifications, pin configurations, and functionalities outlined in the datasheet.

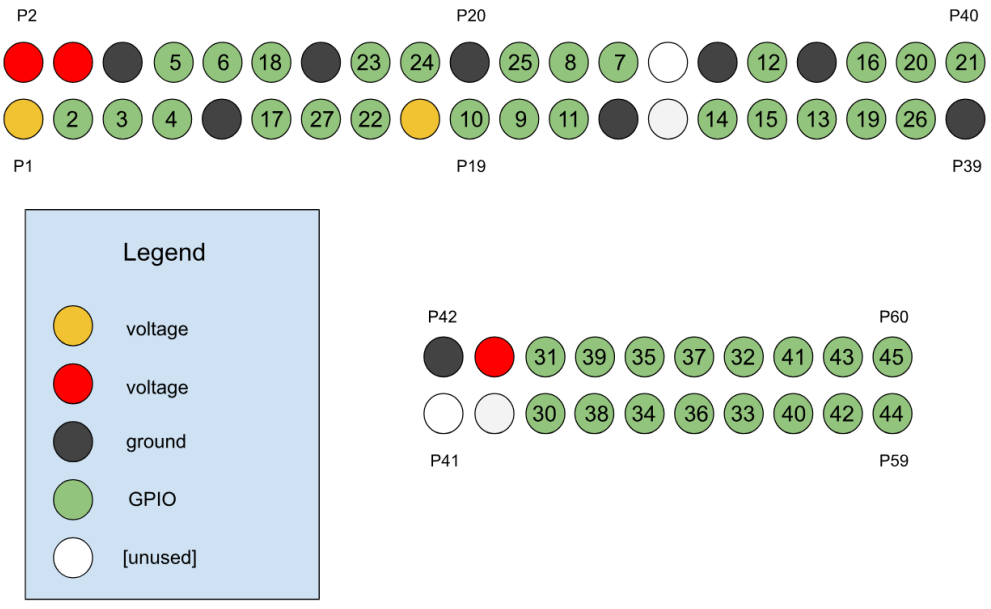
Step 2:
boot_prompt.log:



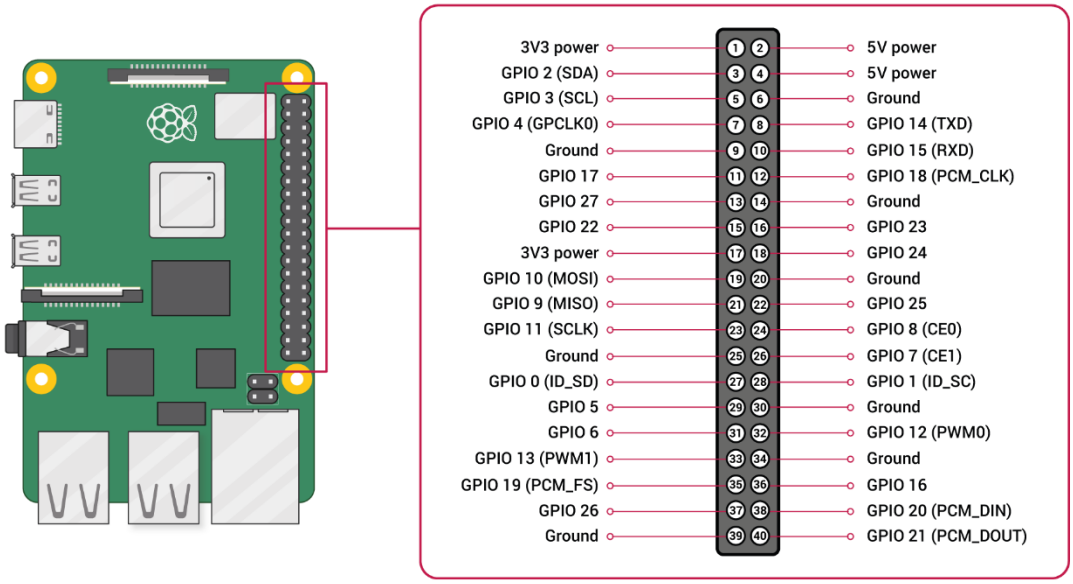
The log file contains information about the start-up process of a collector, which appears to be a device or system that collects data.

The log file shows that the collector was successfully initialized, the configuration was loaded, the services were started, and the collector is now online. The log file also shows that the Alternative Function Assignment is set to ALTO. Evaluated if any information within the boot prompt output provided clues or indications regarding the pins in use or their functionalities.

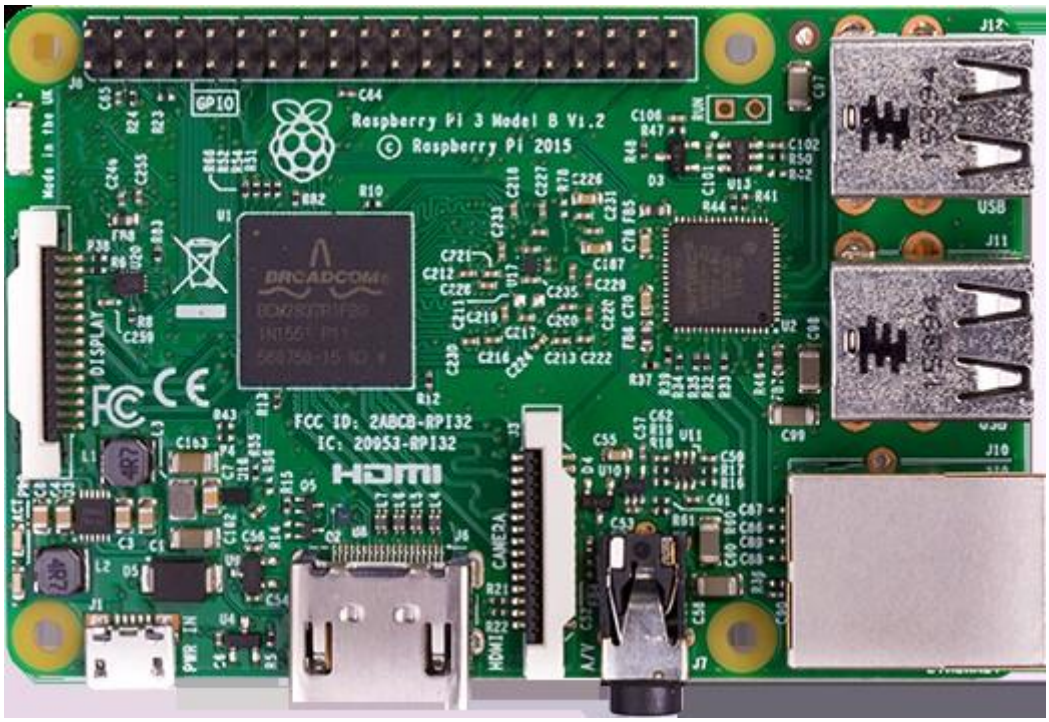
Step 3: pinout.svg



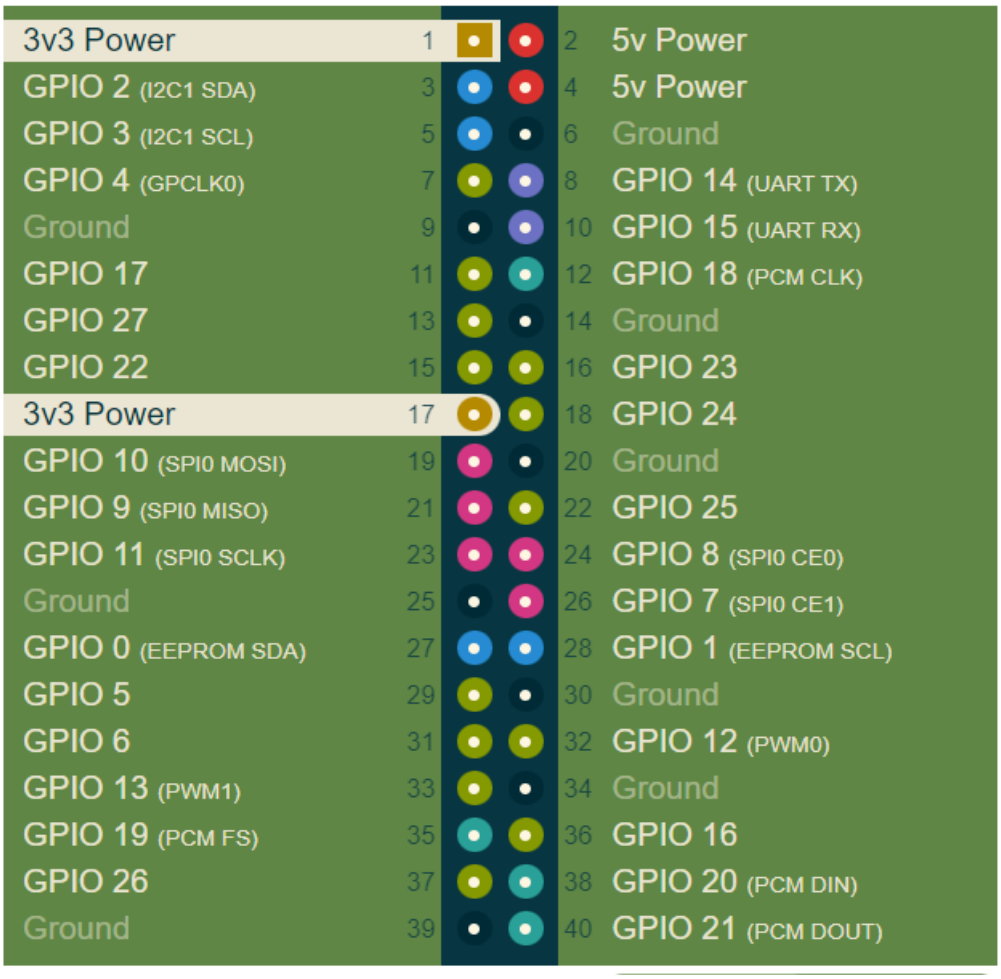
- Examined the provided pinout.svg file, which detailed the physical layout of the device's GPIO and other pins.
- Analyzed the pin labels, identifying the GPIO header, voltage types (3.3v and 5v), and other relevant details.



GPIO and the 40-pin header



Raspberry PI Hardware



Step 4:

Understanding the pinout.svg: Open the rendering of debug ports on the embedded computer (pinout.svg). We need to pay attention to the pins labeled, especially the GPIO (General Purpose Input/Output) header. We have to note the pins associated with power, ground, UART (Universal Asynchronous Receiver-Transmitter) transmit, and receive functions.

PINS Functionality:

Power the GPIO Header: To power the GPIO header, you'll need a pin that supplies voltage to the GPIO pins. Usually, pins marked as 5V or 3.3V on the GPIO header provide power. Typically, pin 2 (5V) or pin 4 (5V) is used to supply power to the GPIO header on a Raspberry Pi.

Ground the Board: Ground pins are essential for completing electrical circuits and providing a stable reference voltage. On the GPIO header, pins marked as GND (ground) are used for this purpose. Common ground pins on a Raspberry Pi are pin 6 (GND) or any pin labeled GND across the header. Additional ground pins are available for better grounding distribution, especially when using multiple components.

UART Transmit Function: UART (Universal Asynchronous Receiver-Transmitter) transmit pins are used to send serial data. On a Raspberry Pi, the UART transmit pin is GPIO14 (TXD), typically found on pin 8 of the GPIO header.

UART Receive Function: UART-received pins are used for receiving serial data. On a Raspberry Pi, the UART receive pin is GPIO15 (RXD), usually located on pin 10 of the GPIO header.

These pins play crucial roles in enabling communication, providing power, and establishing ground connections within the GPIO header of devices like the Raspberry Pi. The specific pins may vary depending on the device's configuration, so always refer to the datasheet or pinout information for precise details.

Once we have access to the necessary resources, we can proceed with the following steps:

1. Locate and analyze the processor datasheet: Identify the pinout diagram and map it to the pinout.svg file for cross-referencing.
2. Analyze the boot_prompt.log: Look for indications of communication protocols (e.g., UART) and potential pin usage.
3. Combine information: Based on the pinout mapping and boot_prompt analysis, identify the specific physical pins enabling data transmission and reception (UART TX, UART RX, etc.).

Step 5:
Cross-reference with Datasheet: Verify the pin locations on the GPIO header in the pinout.svg file with the datasheet's specifications. Identify the physical pins corresponding to the required functionalities (power, ground, UART transmit, and receive).

Step 6:
Solving the Prompts: Based on the datasheet information and the provided rendering (pinout.svg), we will fill in the answers to the prompts.

Step 7:
Verification: Once we've determined the answers based on the datasheet and the provided resources, verify and confirm the solutions by cross-referencing the information.

Achievements:

Hardware identification: Successfully determining the pinout diagram and CPU image to determine the hardware architecture of an unknown device and to provide some insight into its functionality.

Datasheet Utilization: Effectively interpreting technical details from the processor datasheet and converting them to the physical pins allows for a deeper grasp of the device's capabilities.

Pin Function Determination: Correctly identifying which physical PINs are in charge of carrying out particular tasks, such as grounding the board, powering the GPIO header, and locating the send and receive UART operations.

Problem-Solving Proficiency: Exhibiting the ability to solve problems by accurately determining the relevant pin functions by linking data from a variety of sources, such as the CPU image, pinout diagram, and boot prompt output.

Interacting with Firmware: This involves effectively determining low-voltage activity and maybe figuring out how to communicate with the firmware of the device, demonstrating an awareness of data transfer and possible control methods.

Technical Documentation: Providing a thoroughly documented report that describes the analysis's methodology, conclusions, and solutions will guarantee that the functionality and pin configurations of the device are explained with accuracy and clarity.

Adherence to Security Protocols: Ensuring that any interaction with the device's firmware aligns with security protocols and ethical considerations, maintaining the integrity and security of the system.

Resources Used:

Microsoft Word - BCM2835 ARM Peripherals.docx

https://www.raspberrypi.com/documentation/

https://www.raspberrypi.com/documentation/computers/raspberry-pi.html#gpio-and-the-40-pin-header

https://pinout.xyz/pinout/3v3_power

Task 2 - Extract the Firmware - (Hardware analysis, Datasheets) — Points: 100

Thanks to your efforts the USCG discovered the unknown object by trilaterating the geo and timestamp entries of their record with the correlating entries you provided from the NSA databases. Upon discovery, the device appears to be a device with some kind of collection array used for transmitting and receiving. Further visual inspection shows the brains of this device to be reminiscent of a popular hobbyist computer. Common data and visual ports non-responsive; the only exception is a boot prompt output when connecting over HDMI. Most interestingly there is a 40pin GPIO header with an additional 20pin header. Many of these physical pins show low-voltage activity which indicate data may be enabled. There may be a way to still interact with the device firmware...

Find the correct processor datasheet, and then use it and the resources provided to enter which physical pins enable data to and from this device

Hints:

- Note: For the pinout.svg, turn off your application's dark mode if you're unable to see the physical pin labels (eg: 'P1', 'P60')
- The pinout.svg has two voltage types. The gold/tan is 3.3v, the red is 5v.
- The only additional resource you will need is the datasheet, or at least the relevant information from it

Downloads:

- Rendering of debug ports on embedded computer (pinout.svg)
- image of device CPU (cpu.jpg)
- copy of display output when attempting to read from HDMI (boot_prompt.log)

Prompts:

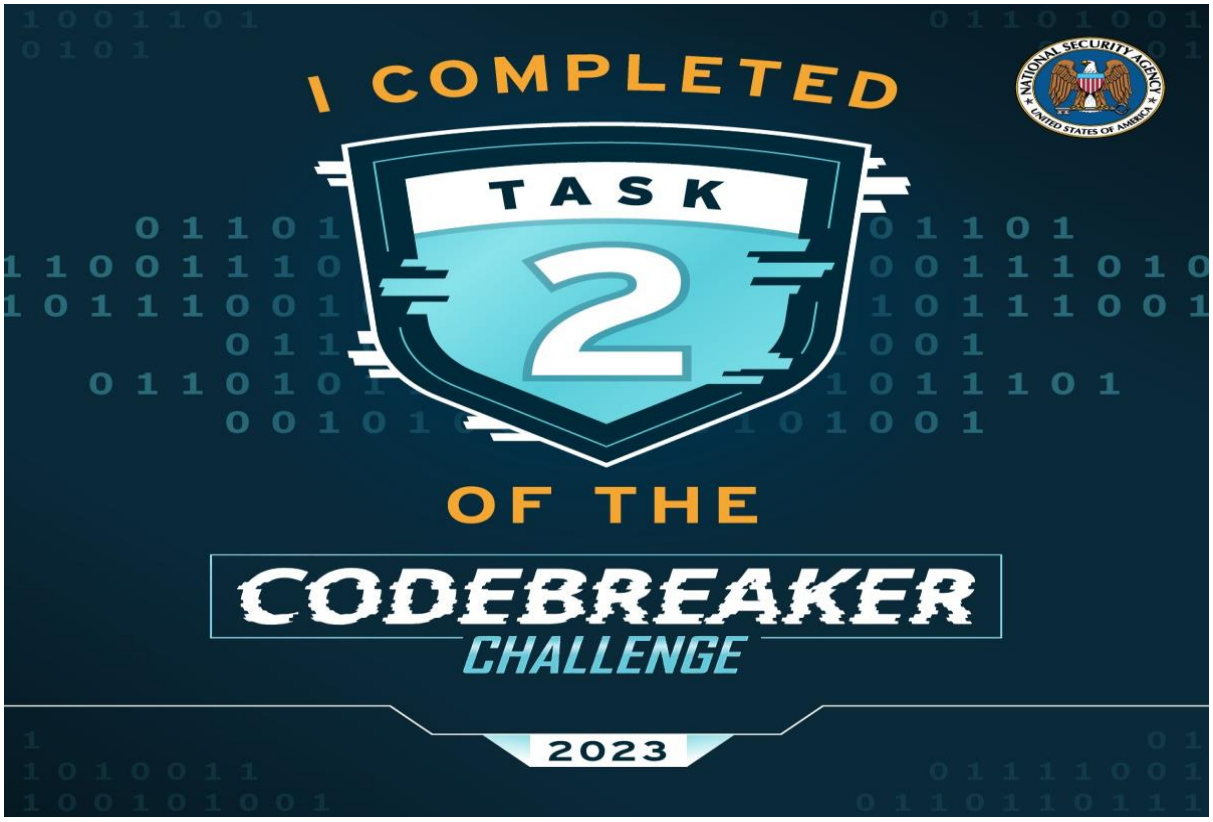
- Provide the correct physical pin number to power the GPIO header
- Provide a correct physical pin number to ground the board:
- Provide the correct physical pin number for a UART transmit function:
- Provide the correct physical pin number for a UART receive function:

Task Completed at Tue, 05 Dec 2023 23:25:46 GMT:

Well done! All GPIO functions map to the correct physical ports of the headers. We can now interact with the device over UART!

Downloads:

- Task 2 Badge (badge2.png)



Conclusion:

The task focused on identifying specific physical pins on an embedded device marked by the BCM2837RIFBG CPU. It required deciphering functionalities to power the GPIO header, ground the board, and locate UART transmit and receive function pins using available resources. Researching the CPU datasheet, analyzing the provided pinout.svg and boot prompt output, and cross-referencing data facilitated the identification process. This challenge highlighted the importance of meticulous analysis and resource integration in hardware-oriented problem-solving within computing systems.