HARDWARE Find the Datasheet

Goals of this challenge

- Identifying the microcontroller
- Flashing the firmware
- Solving the challenge

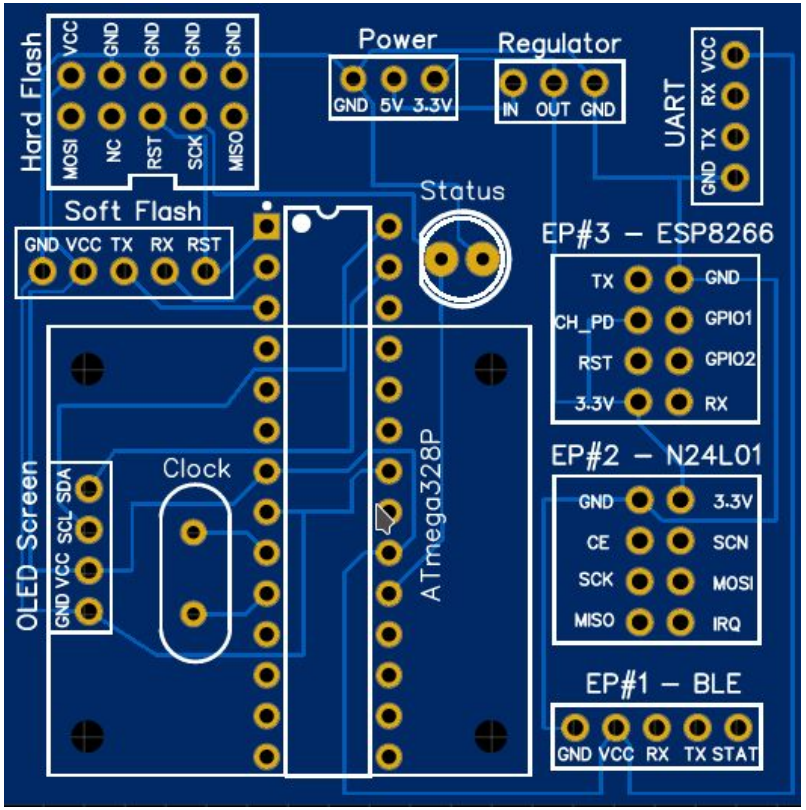
Challenge Description: **Read the password transmitted on the microcontroller pin PD1 of the DVID board**

Identifying the microcontroller

- Inspect the Gerber files in the DVID/build directory [github]
- Find the board schematics file or use gerber viewer to get a jpeg for analysis
- Identify the MCU on the board

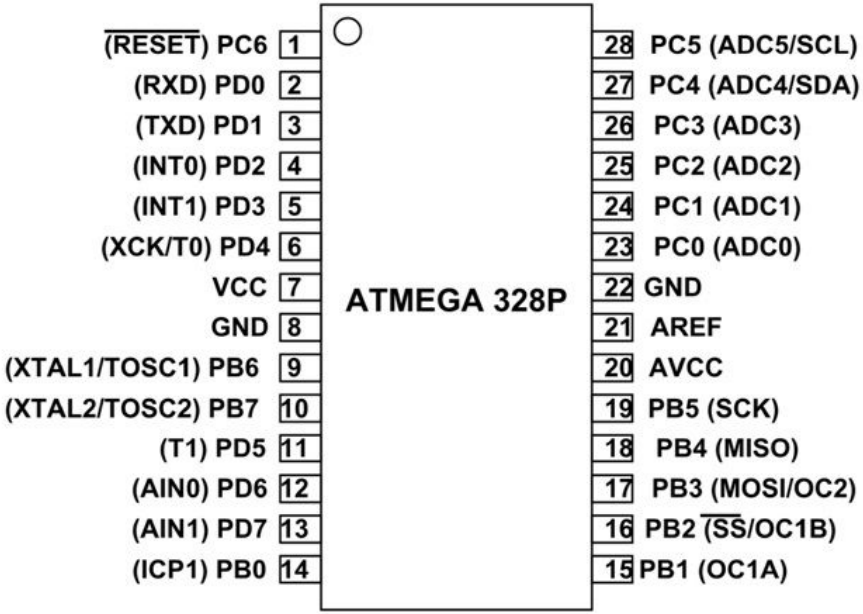
Hint: The MCU uses a DIP socket

Bonus: Also identify other components and download the datasheets



Identifying the microcontroller

- On the board schematics, we can see a **PDIP socket** for an **ATmega328p** chip
- From the datasheet we know that it is an **8-bit AVR** Microcontroller with **32K** Bytes In-System **Programmable Flash**
- **PD1** port is located on the **third port**, starting at top left of the chip.
- This pin is connected to **Custom Port #3** on the DVID board



Note:

ATmega328p_datasheet.pdf
https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

Now that we have identified the port used to transmit the flag, we can now learn how to flash the firmware

Flashing the firmware

- In order to do this, we will use **avrdude** and an **USB AVR programmer.**
- Use the **flash.sh** or flash.bat file for flashing the firmware

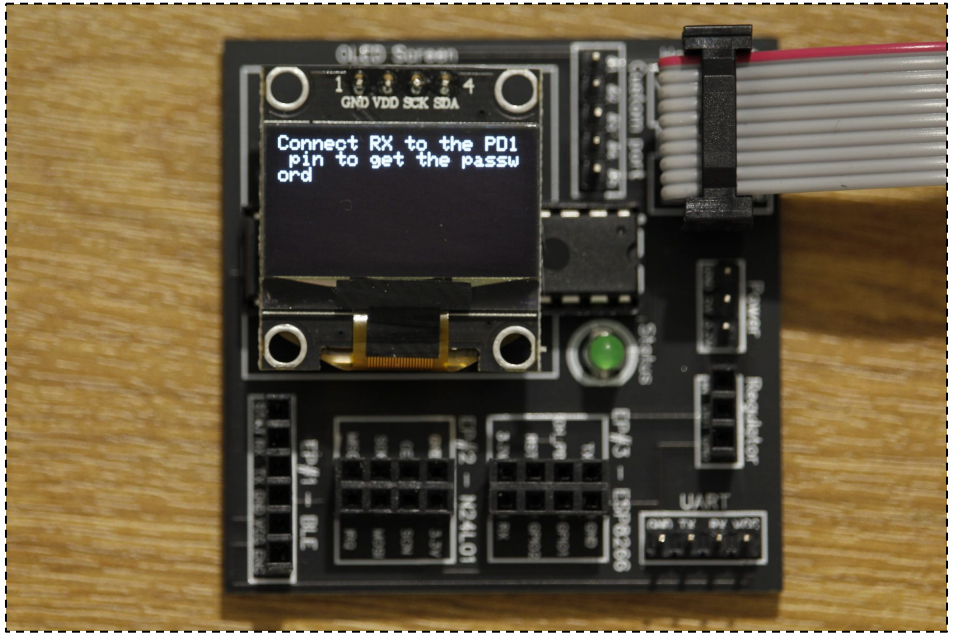
avrdude **options**	
-F	Override invalid signature check.
-v	Verbose mode
-p atmega328p	Target microcontroller type, here we have an ATmega328p
-P /dev/ttyUSB0	Specify connection port, in this case the device /dev/ttyUSB0
-c usbasp	Specify programmer type to use.
-u	Disable safemode, default when running from a script.
-U flash:w:firmware.hex	Memory operation specification <memtype>:r|w|v:<filename>[:format]. Here we write (w) on the flash the file firmware.hex

Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip
Flash.sh
https://hastebin.skyra.pw/raw/likojafomo

Flashing the firmware

- Connect the **DVID board** to the **computer** using the **USB AVR Programmer**, and start the script
- When the AVR programming has completed, the **board should restart** and you should the following

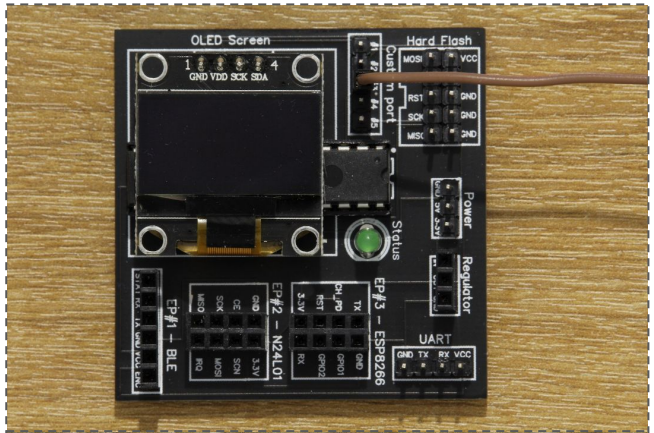


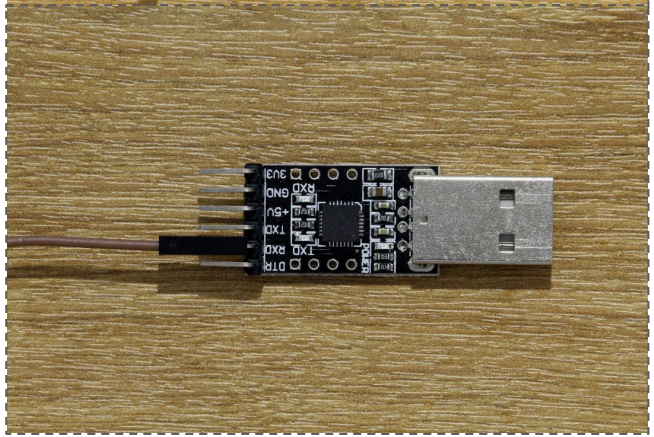
Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip
Flash.sh
https://hastebin.skyra.pw/raw/likojafomo

Solving the challenge

- In order to read the **password** we will connect a **USB UART reader RX pin** to **PD1 pin** on the **chip**
- As we have seen before, We will **connect RX pin** of the **USB UART reader** to **custom port #3**
- In order to **read the data** coming from the **USB UART reader** we can either write a **python script**, or use tools like **minicom**. We'll try with a classic bitrate of 9600 bps.





Note:

script.py
https://hastebin.skyra.pw/raw/jizuzecini

Solving the challenge

- Python Script
- Using minicom

```python
#!/usr/bin/env python3

import serial

s = serial.Serial("/dev/ttyUSB0", 9600)
while True:
    print(s.read())
```

```
minicom -D /dev/ttyUSB0 -b 9600
```

Note:

script.py
https://hastebin.skyra.pw/raw/jizuzecini

Solving the challenge

DVID:findTheDatasheet

```
Welcome to minicom 2.7.1

OPTIONS: I18n
Compiled on Dec 23 2019, 02:06:26.
Port /dev/ttyUSB0, 10:42:58

Press CTRL-A Z for help on special keys
```

Hint:

One can simply use the serial monitor tool from Arduino IDE