ESP32 Pulse Analysis Toolkit

Setup Guide

1 Initial Setup

1.1 Install Arduino IDE

1. Download Arduino IDE from `https://www.arduino.cc/en/software`
2. Install the software following the instructions for your operating system
3. Launch Arduino IDE

1.2 Install ESP32 Board Support

1. In Arduino IDE, go to **File** → **Preferences**
2. In ”Additional Boards Manager URLs” field, add:
   `https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_`
3. Click **OK**
4. Go to **Tools** → **Board** → **Boards Manager**
5. Search for ”esp32”
6. Find ”ESP32 by Espressif Systems” and click **Install**
7. Wait for installation to complete

1.3 Install USB Drivers

1.3.1 Windows

- For CP210x-based boards: Download from `https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers`
- For CH340-based boards: Download from `https://www.wch.cn/download/CH341SER_EXE.html`
- Run the installer and follow instructions

1.3.2 macOS

- For CP210x-based boards: Download from `https://www.silabs.com/developers/usb-to-uart-bridge-vcp-drivers`
- For CH340-based boards: Download from `https://www.wch.cn/download/CH341SER_MAC_ZIP.html`
- Install and restart if needed

1.3.3 Linux

```
sudo usermod -a -G dialout $USER
sudo usermod -a -G tty $USER
# Log out and log back in for changes to take effect
```

1.4 Install Python Dependencies

```
pip install -r requirements.txt
```

2 Hardware Setup

2.1 Circuit Diagram

```
VCC (Programmed Potential) ----[470 ohm]----+----[Device Under Test]----[470 ohm]----
                              |
                              |
                       ESP32 ADC Pin (GPIO34)
```

2.2 Physical Connection Steps

1. Connect a 470 ohm resistor from VCC (3.3V) to one terminal of your device under test

2. Connect another 470 ohm resistor from the other terminal to ground

3. Connect ESP32's ADC pin (GPIO34) to the junction between the first resistor and your device

4. Make sure ESP32 and your circuit share a common ground

5. Connect ESP32 to computer via USB cable

3 Uploading Sketches

3.1 For Pulse Detection and Power Analysis

1. Open Arduino IDE

2. Open `pulse_detector.ino`

3. Select your ESP32 board: **Tools** → **Board** → **ESP32 Arduino** → **ESP32 Dev Module**

4. Select correct port: **Tools** → **Port** → *[Your ESP32 port]*

5. Click the **Upload** button (right arrow icon)

6. Wait for "Done uploading" message

3.2 For Waveform Analysis

1. Open Arduino IDE
2. Open `waveform_capture.ino`
3. Select your ESP32 board (as above)
4. Select correct port (as above)
5. Click **Upload**
6. Wait for ”Done uploading” message

4 Running Analysis Scripts

4.1 For Power Analysis

```
python power_analysis.py
```

Follow the prompts to:

- Select serial port
- Enter pulse voltage (e.g., 3.3V)
- Enter resistor value (default: 470 ohm)
- Choose recording duration

4.2 For Pulse Detection

```
python pulse_detector.py
```

Follow the prompts to configure session parameters.

4.3 For Waveform Analysis

```
python waveform_detector.py
```

Follow the prompts to configure session parameters.

5 Troubleshooting

- **Cannot find port**: Check cable connection and make sure drivers are installed
- **Upload fails**: Hold BOOT button while uploading if using ESP32 DevKit
- **No readings**: Check circuit connections and resistor values
- **No data received**: Verify baud rate (115200) matches in Arduino sketch and Python script