Rotating Display RD40
Shift Register Timing Analysis

1. **Preparation of the board**
   Before you can start the analysis with the oscilloscope, you need to prepare your board by soldering wires or pins to the points of interest of the Arduino nano.

   We will look at

SPI Clock	GPIO 13
SPI MOSI (Data)	GPIO 11
G_enable	GPIO 10
Latch	GPIO 9

   It is probably easiest, to solder pins to the top of the Arduino header:

2. **Checking the ripple of the +5V line**
   Start with connecting the oscilloscope to the +5V line. A good access point is Resistor R2. Connect the ground clip to the ground pin of one of the diodes:

Use channel 1 of the oscilloscope.

Settings:
Trigger: channel 1
AC coupling
Amplification channel 1: 100 mV/cm
Time base: 4μs/cm
Adjust the trigger level until you get a stable image.

You can expect to see some noise with the frequency of the wireless power supply (about 120 kHz, period 8 μs).

This is what I got on my board:

The amplitude of the ripple on my board is about 100 mV, which is very good. Measure the ripple on your board.

3. **Checking the SPI Clock**
   Connect your scope to GPIO 13.

Settings:
Trigger: channel 1, rising edge
DC coupling
Amplification channel 1: 2V/cm or 5V/cm
Time base: 4µs/cm
Adjust the trigger level, until you get an image. The image will not be stable, as the signal is irregular. So it might be helpful, to simply stop triggering at some point and in order to do an image analysis. My scope has a button for that.

This is how the signal should look:

There are 5 bursts with 8 pulses each. These are the five bytes of data for the 40 LEDs (one byte per shift register). There is a reason, why there is a longer pause

between the second and the third burst. As you can see in the Arduino code, the third byte needs more computation time.

Next we check the clock frequency.

Settings:
Trigger: channel 1, rising edge
DC coupling
Amplification channel 1: 2V/cm or 5V/cm
Time base: 400ns/cm

Measure the time of 1 period. In my case, it is 1 µs, which corresponds to 1MHz. This is the SPI clock frequency set in the (newly modified) Arduino nano program.

4. **The Latch signal**
   A rising edge of the Latch signal causes the shift register to transfer the previously send data to the LED outputs. In other words: there needs to be a rising edge, once the data transfer to the shift registers is complete.

   Connect channel 1 of the oscilloscope to clock, channel 2 to Latch:

Settings:
Trigger: channel 1, rising edge
Channel 1 DC coupling
Amplification channel 1: 2V/cm or 5V/cm
Channel 2 DC coupling
Amplification channel 2: 2V/cm or 5V/cm
Time base: 4µs/cm

This is how the scope should look like. Note the rising edge of Latch after the 5 clock bursts.

5. **The STI data line (MOSI)**
   Connect channel 1 to CLK, channel 2 to SPI MOSI:

Settings (same as previous):
Trigger: channel 1, rising edge
Channel 1 DC coupling
Amplification channel 1: 2V/cm or 5V/cm
Channel 2 DC coupling
Amplification channel 2: 2V/cm or 5V/cm
Time base: 4µs/cm

The image will depend on whether the ESP01 sends data, or not. If so, the data line will be different from one scope image to the next. Here is a screen shot with scope on “STOP”:

6. **G enable signal**
   This signal turns on and off all LEDs. The LEDs are on, when the signal is LOW. The length of the signal is used to control the brightness of the display. It is controlled by timer 1.

   Connect channel 1 to CLK, channel 2 to G_enable.

Settings:
Settings (same as previous):
Trigger: **channel 2**, rising edge
Channel 1 DC coupling
Amplification channel 1: 2V/cm or 5V/cm
Channel 2 DC coupling
Amplification channel 2: 2V/cm or 5V/cm
Time base: 8µs/cm

As you can see, the LEDs are off most of the time. Shortly after the LEDs are turned off, the data transfer of the next LED row starts.