

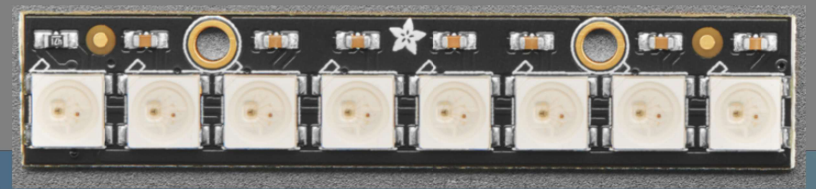
R5 Control of NeoPixel



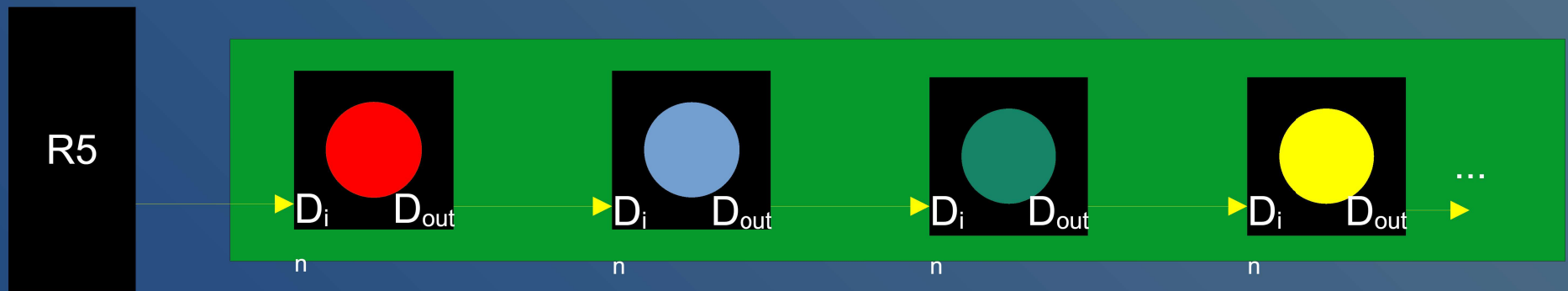
Topics

- 1) How can we drive **RGBW** LEDs (**NeoPixel**) from the BeagleY-AI?

About the NeoPixel

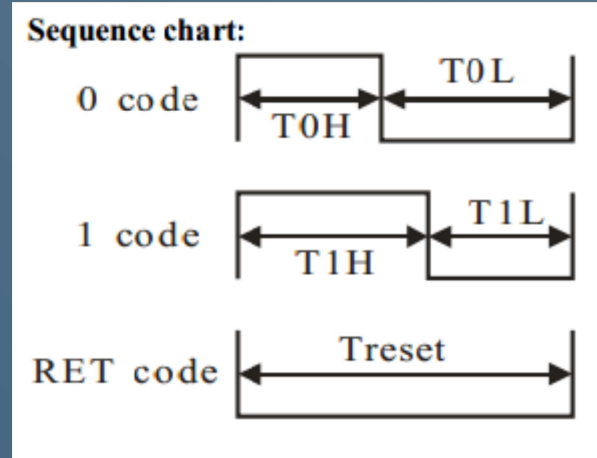


- Adafruit makes RGB LEDs called NeoPixel
 - Uses a 1-wire protocol
(Features WS2812B or SK6812 LED driver)
 - Pixels have D_{in} and D_{out} daisy chained:
As data is shifted into a pixel,
it simultaneously shifts old data.
 - Designed for 5V, but works on 3.3V!
 D_{in} signal can be $0.7V_{DD} = 0.7 \cdot 5 = 3.5V$ (but it works!)

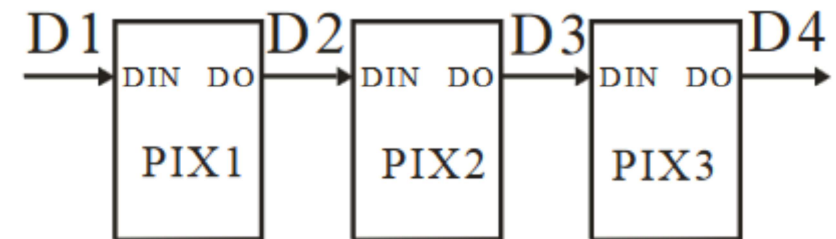


1-Wire Protocol

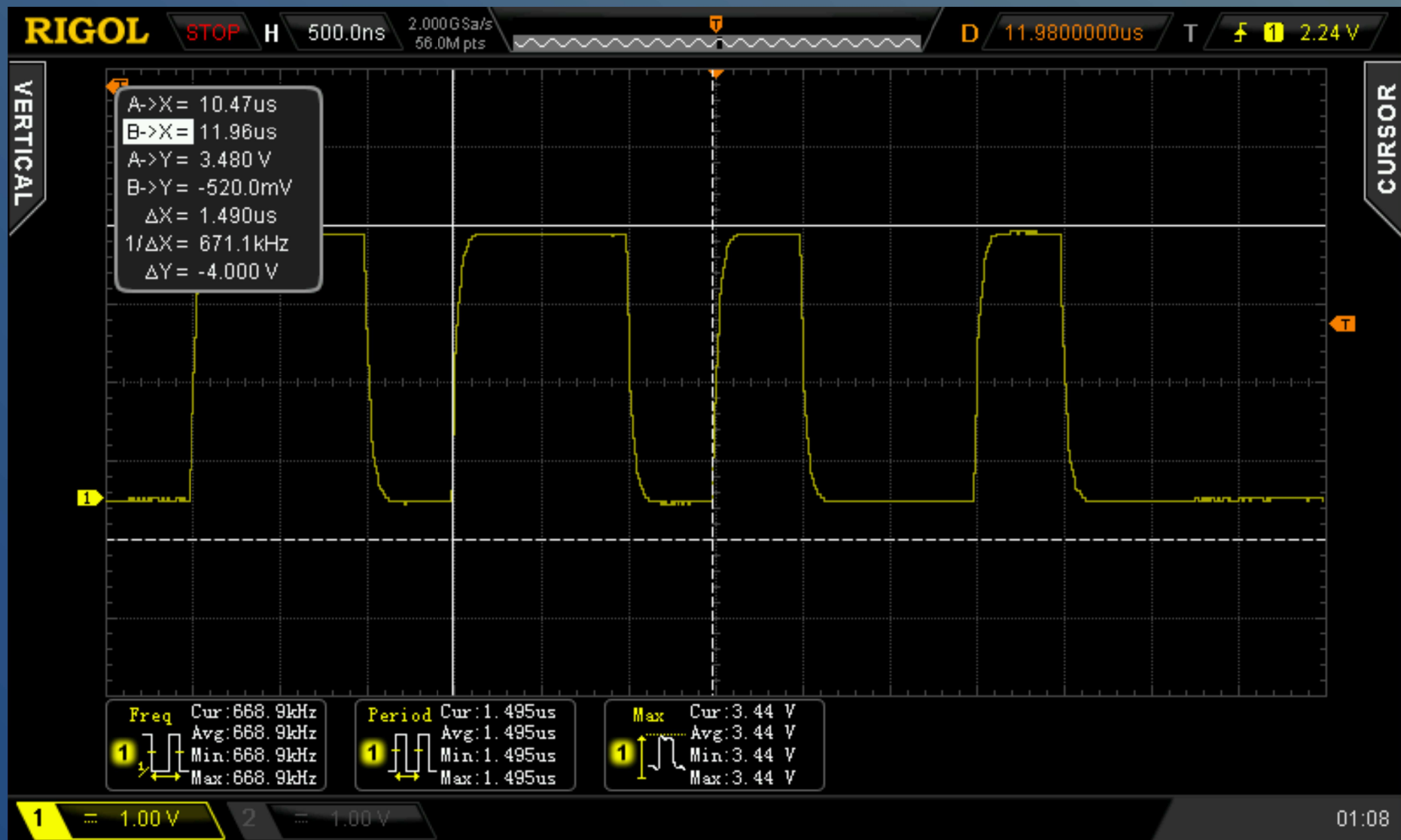
- The Data wire signal is a square wave
 - 0: ..
(0.35us = 350ns)
 - 1: ..
(0.7us = 700ns)
- At 800 MHz, how many clock cycles is 0.35us?
 - $1s / 800 \text{ Million Cycles/s}$
 $= 0.00125 \text{ us / Cycle}$
= 1 R5 CPU cycle takes 1.25ns
 - Note 0.35us = 350ns
 - # cycles in 350ns
 $= 350 \text{ ns} / 1.25 \text{ ns/cycle}$
= 280 cycles



Cascade method:

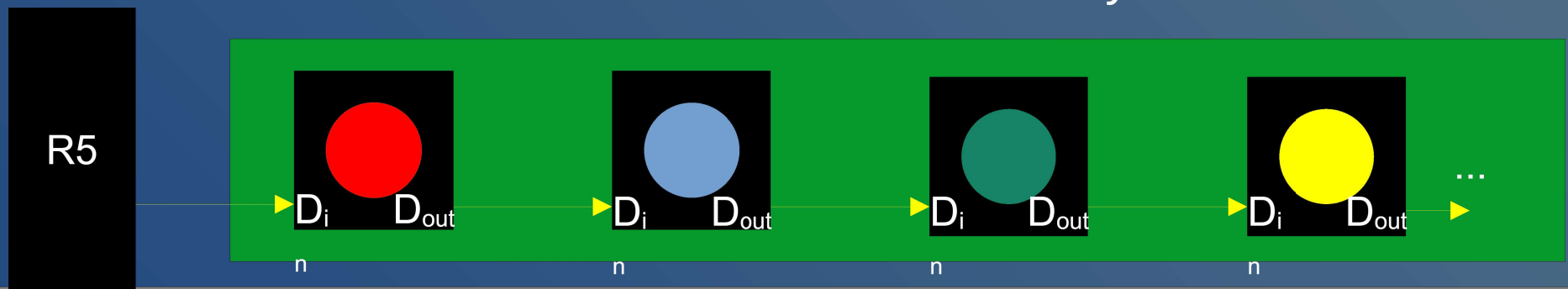


Actual Wave



Frame

- Send a single “frame” to show on the LEDs (RGBW values for all pixels)
 - Shift in all bits, for all pixels, one at a time.
 - Send the last LED’s values first (shifts through all)
 - Send data in the sequence R, G, B, W (8-bit each)
 - Send the high-bit first
- After sent whole frame, signal a RESET
 - Pull data line low for $\geq 50\mu s$
 - LEDs don’t show their new colour until they see RESET



Wiring Up



Review Questions

- How is **one wire** used to **send data**?
- How **many pulses** are needed to drive **8 RGBW LEDs**?
- What is the **purpose** of holding the **data line low for >50us**?
- Links
 - [NeoPixel Parts](#)
 - [Data sheet](#)
 - [Info on using NeoPixel](#)