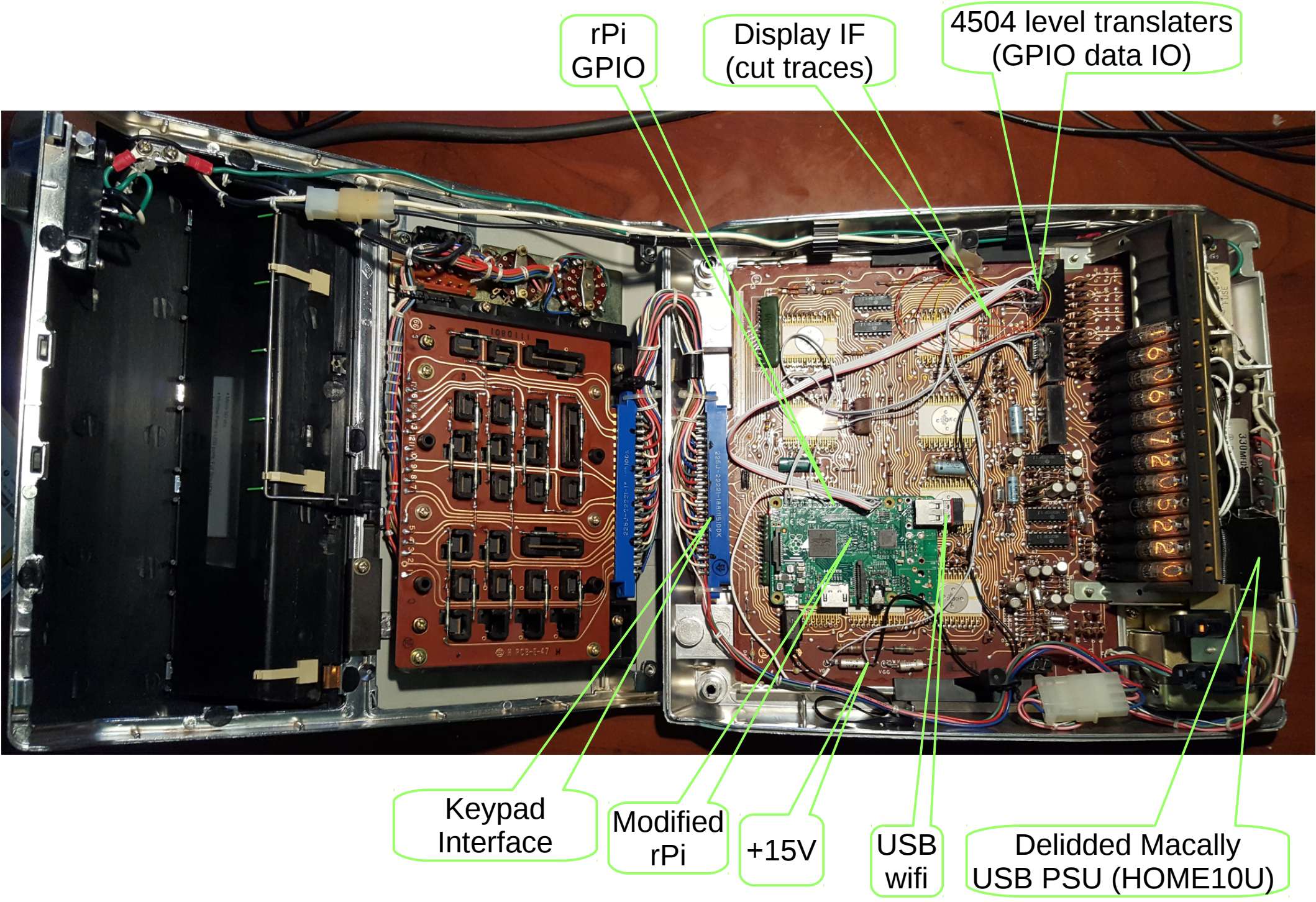

0.1234567898 7
Friden EC 1117
SINGER
SAMSUNG
Nixie
SINGER
Friden EC 1117

Introduction

- Create programmable clock from Nixie tube calculator
  - Reverse engineer display interface using oscilloscope
    - No schematics and no datasheets available whatsoever
    - Figure out correct interface points on PCB (cut traces)
    - Interface Raspberry Pi to vintage high voltage electronics
    - Write C code to emulate display protocol in software
      - Runs UDP server thread to handle remote requests via wifi
  - Remote control from Android phone over wifi
    - Write Java Android app to control rPi C code
      - UDP client; simple ASCII protocol
    - Support switching between multiple date/time display formats
    - Support setting individual digits via wheel controls
- Created by: Eric D. Cohen <nixie@epieye.com>, 2016

Hardware

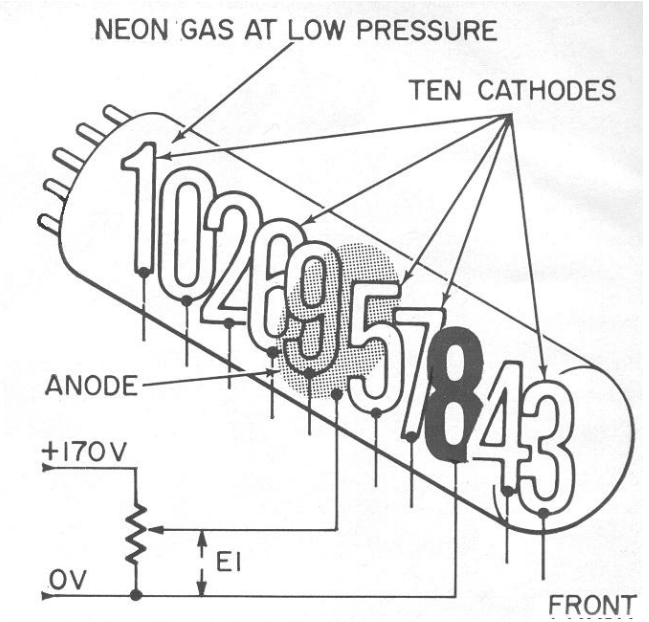

rPi
GPIO
Display IF
(cut traces)
4504 level translaters
(GPIO data IO)
Keypad
Interface
Modified
rPi
+15V
USB
wifi
Delidded Macally
USB PSU (HOME10U)

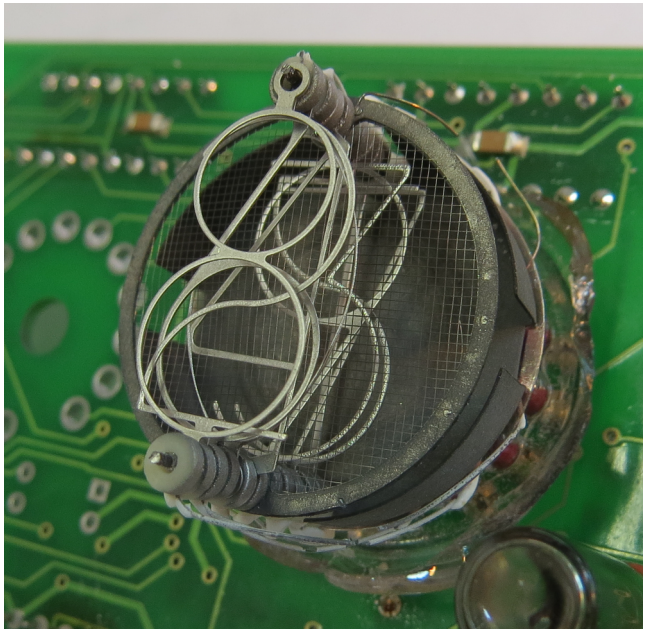
Singer-Friden-Hitachi EC1117

- Founded by Carl Friden in San Leandro, CA, 1934
  - R&D center in Oakland, CA
  - Initial products were typewriters and mechanical calculators
  - Bought by Singer (Sewing Machines!) in 1965
- EC1117 released in 1971
  - First to use LSI ICs
  - Retail price of $445 ($2,900 today)
  - 12 digits in Hitachi CD-90 Nixie tubes (14 digits for EC1118)
    - EC1117A→cost reduced version using early VFD tubes
  - Fixed point with interesting additional arithmetic features
  - Extremely robust magnetic reed switch keypad

Nixie Tubes

- Haydu/Borroughs circa 1955
  - **N**umeric **I**ndicator e**X**perimental Number **1** (NIX 1→Nixie)
- Cold cathode
  - No filament like traditional tube
  - Each digit a cathode
- Operates around 200V@2mA
  - Switching circuit grounds desired cathode (digit)
  - Off digits float (avoid leakage)



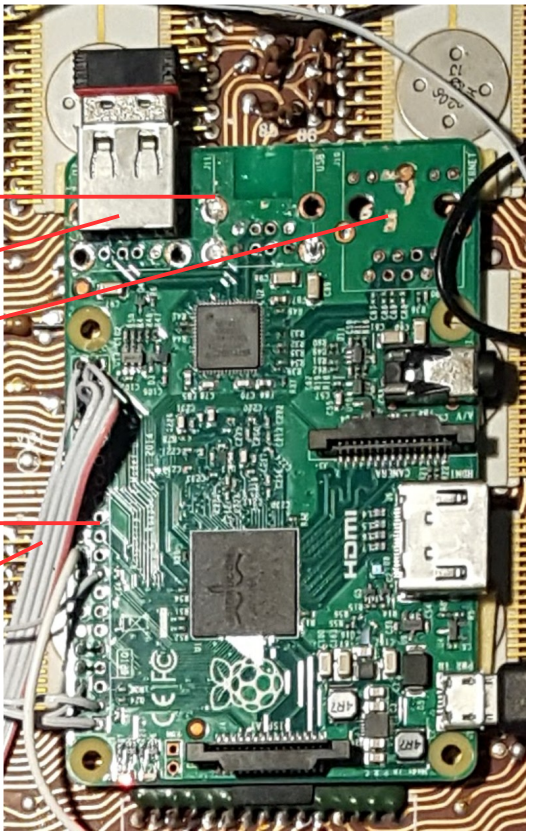


Raspberry Pi 2B (a21041)

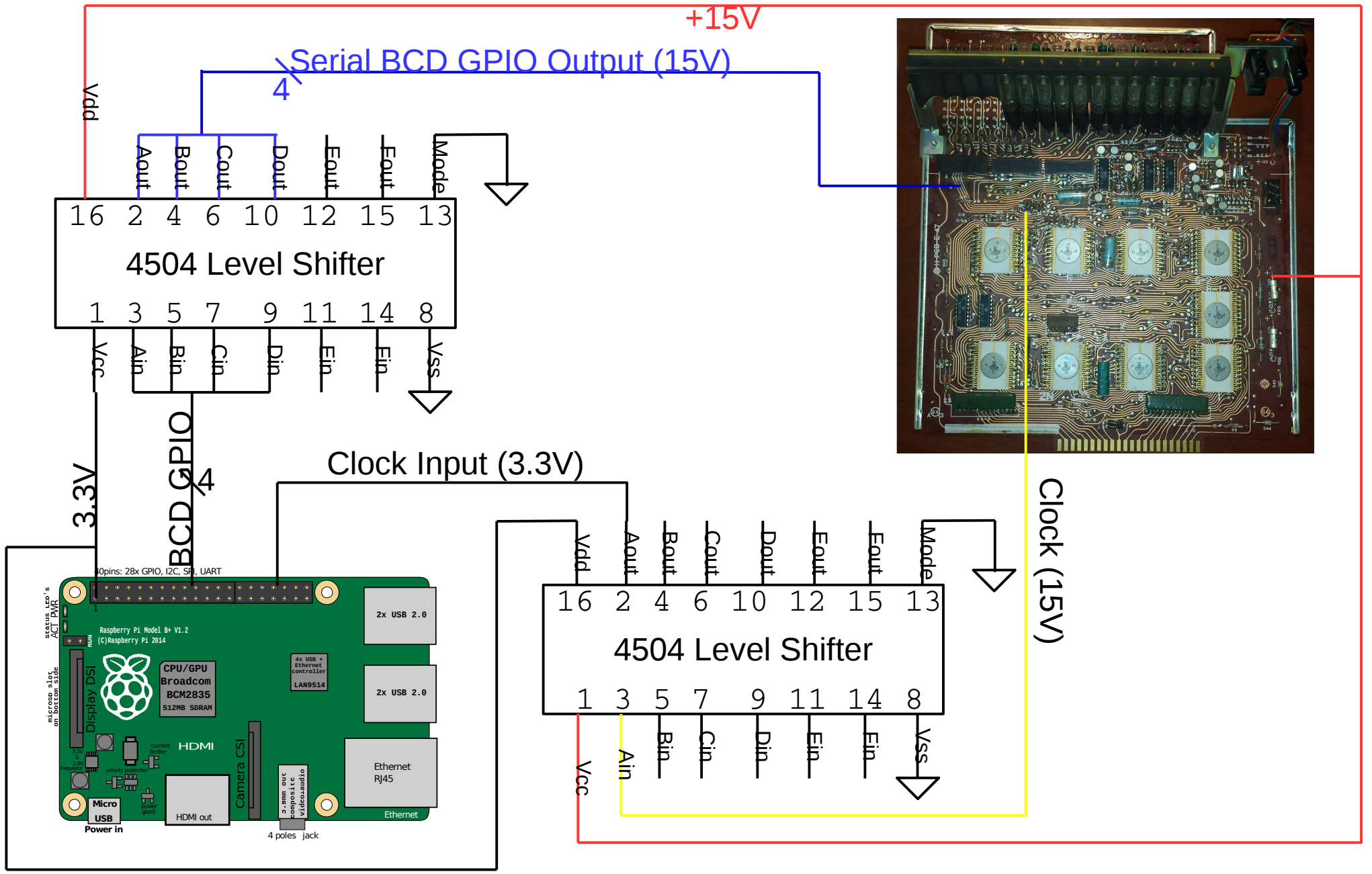
- Cheap: $35
- Powerful: quad core 900MHz Cortex-A7
  - 1GB RAM
- Runs reasonably cool without heatsink
- Many configurable GPIO pins
- Mature Linux distros
- Compact enough

Packaging (rPi Board Rework)

- Limited headroom in chassis
  - Desolder both dual USB ports
    - Replace with single USB port for wifi
  - Desolder Ethernet RJ45 port
  - Desolder all GPIO headers
    - Directly solder ribbon cables for GPIO
- Attach directly to calculator PCB with foam mounting tape

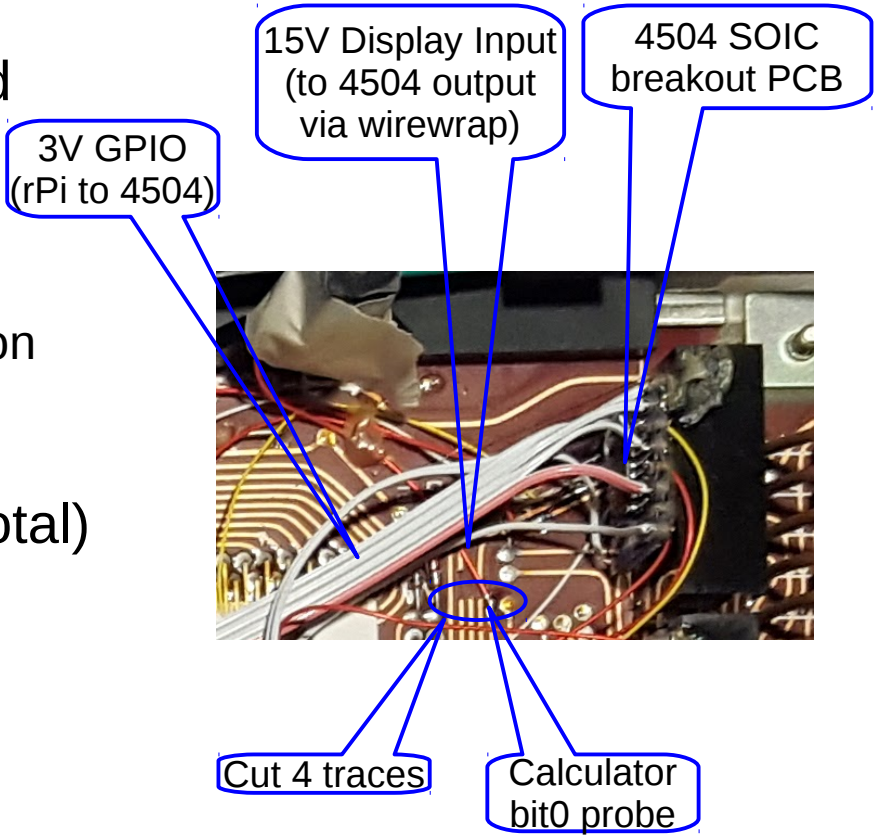


Notional Schematic



Display Bus

- 4 bit serial BCD
  - 12 digits across 4 lines time multiplexed
- Active low
  - 15 volt logic
    - Use 4504 level shifters for for 3V conversion
- 60uS bit time
  - 16.7 kb/s data rate per line (66.7 kb/s total)
- 860Hz system clock (***how quaint!!***!)
  - 1,163uS system clock period
- 115uS high preamble
- Seems to have optional stop bit



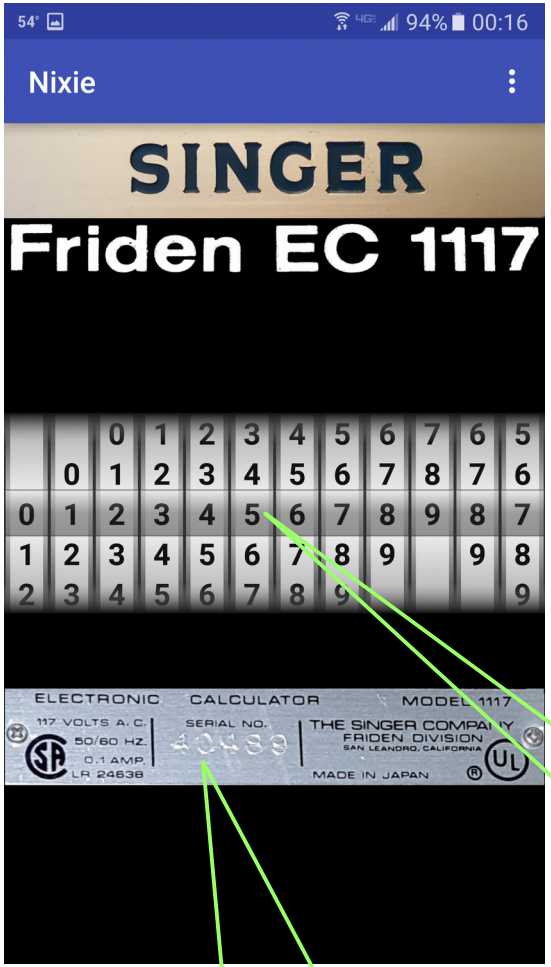
Software

Raspberry Pi C Code

- rPi runs Linux/Raspbian
  - C code runs as standard user process; no RTOS
    - Sometimes misses RT deadline; probably scheduler IRQ
    - Depends on WiringPi for GPIO support
    - Poll GPIO in tight loop for system clock rising edge
      - Wait 115uS after rising edge
      - Drive all 4 data lines for the first digit; hold (spin) for 60uS
      - Drive all 4 data lines for the second digit; hold for 60uS
      - Repeat until all 12 digits have been driven 60uS each
      - Poll for next rising edge; repeat from top
    - Always poll/spin: any context switch → missed deadlines
    - Open loop within system clock period → jitter
      - 19 bittimes within each system clock cycle
  - Listen for UDP packet for mode change, digit program
    - Lockless to avoid timing issues

Android App

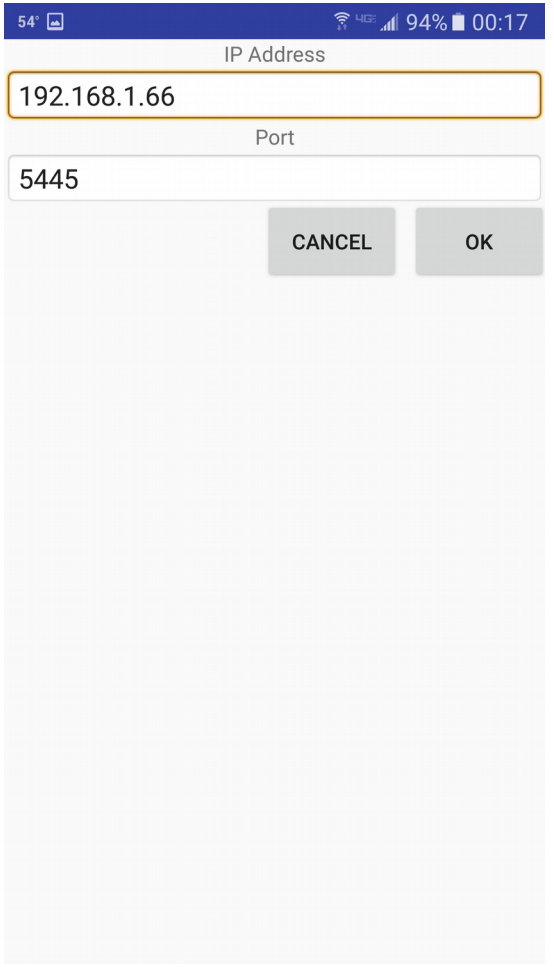
Main Activity



Real-time individual digit control

Swipe right → next mode
Swipe left → previous mode

Configuration Activity



Android App

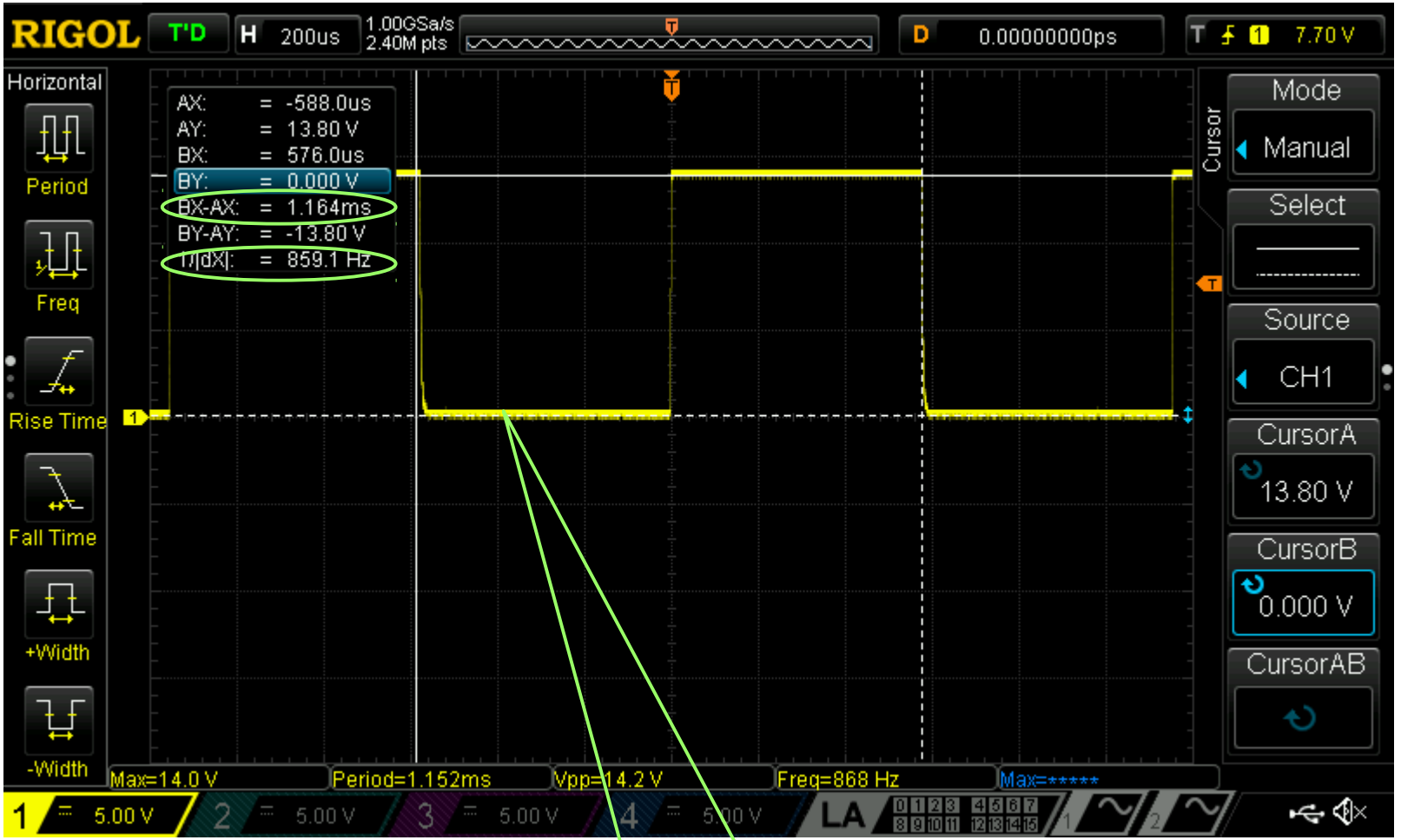
- My very first Android app
  - Depends on android-wheel library
  - Back panel nameplate image controls mode
    - Swipe right or left for next or previous mode
    - Date/time formats, counter mode, digit control mode, etc
  - Emits 14-byte ASCII control message for UDP send
    - ASCII message Format: `<op>,<12 digits>`
    - Digits may be ignored depending on op, but must be sent
    - Ops: 0→next mode, 1→previous mode, 2→digit control

Waveform Walkthrough

Waveform Walkthrough

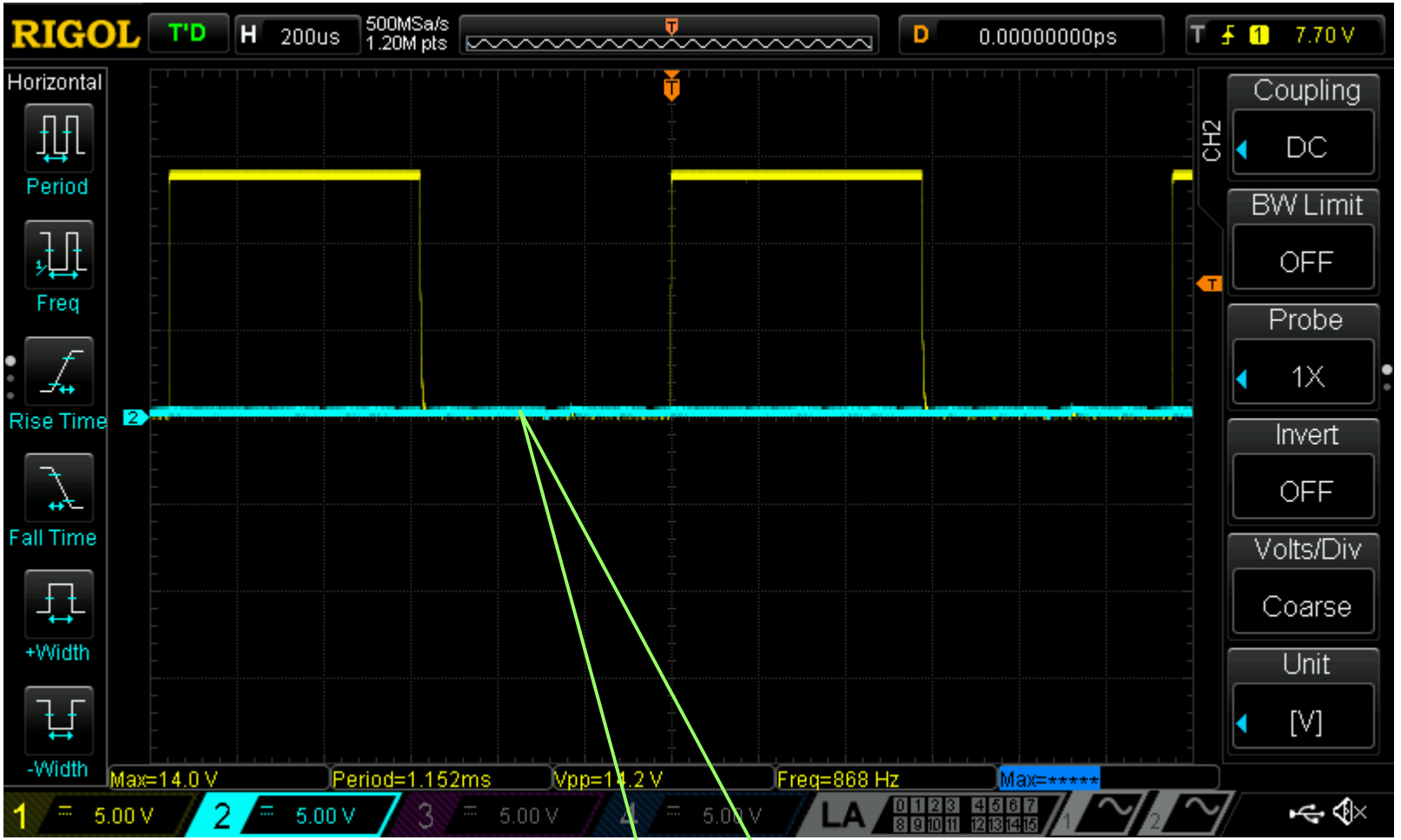
- Oscilloscope captures show only bit zero
  - Remaining three bits have same protocol
  - Digit formed by simultaneously sampling all four bits at given time offset
  - Digit n center time offset (uS): T(n) = 60n + 145
- Yellow trace: 860Hz system clock
- Blue trace: calculator (native) generated bit zero
- Purple trace: emulated (rPi) bit zero

860Hz Clock

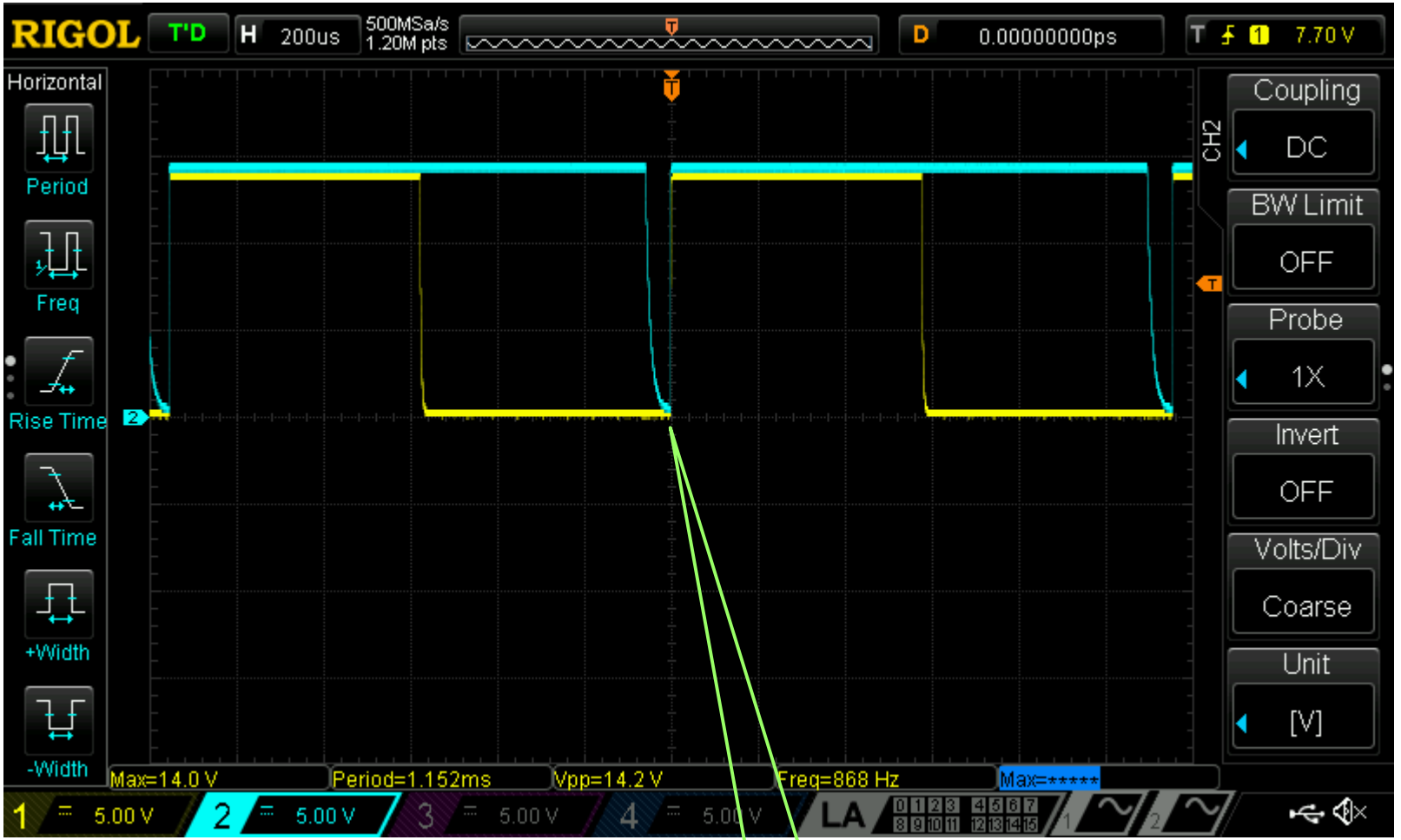


860Hz system clock

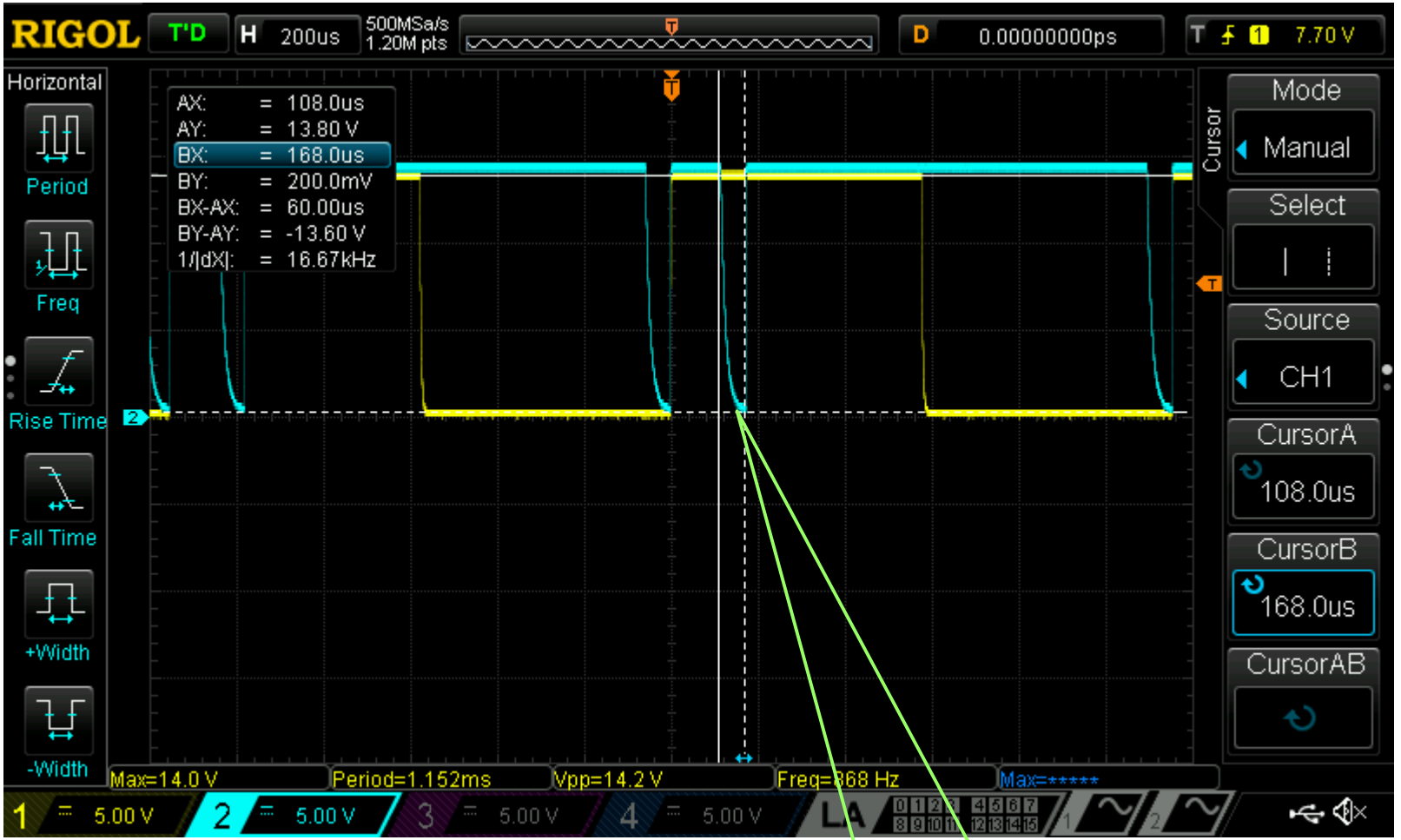
Pre-Clear


RIGOL
T'D
H 200us
500MSa/s
1.20M pts
D 0.00000000ps
T 7.70 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
CH2
Coupling
DC
BW Limit
OFF
Probe
1X
Invert
OFF
Volts/Div
Coarse
Unit
[V]
Max=14.0 V
Period=1.152ms
Vpp=14.2 V
Freq=868 Hz
Max=*****
1 5.00 V
2 5.00 V
3 5.00 V
4 5.00 V
LA
Native display signal
bit 0

Post-Clear

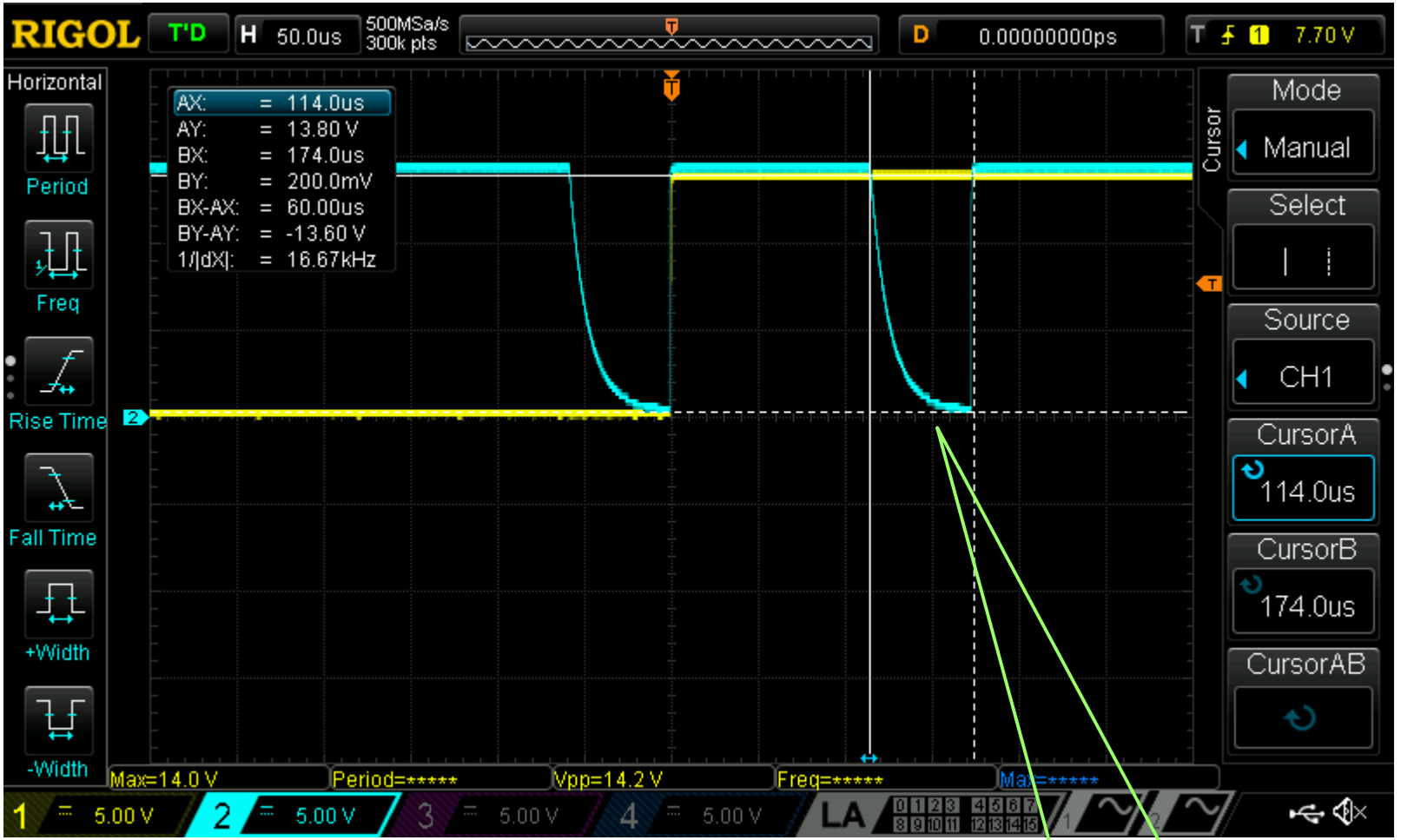

RIGOL
T'D
H 200us
500MSa/s
1.20M pts
D 0.00000000ps
T 7.70 V
Horizontal
Period
Freq
Rise Time
Fall Time
+Width
-Width
CH2
Coupling
DC
BW Limit
OFF
Probe
1X
Invert
OFF
Volts/Div
Coarse
Unit
[V]
Max=14.0 V
Period=1.152ms
Vpp=14.2 V
Freq=868 Hz
Max=*****
1 5.00 V
2 5.00 V
3 5.00 V
4 5.00 V
LA
Stop bit

Display 000000000001



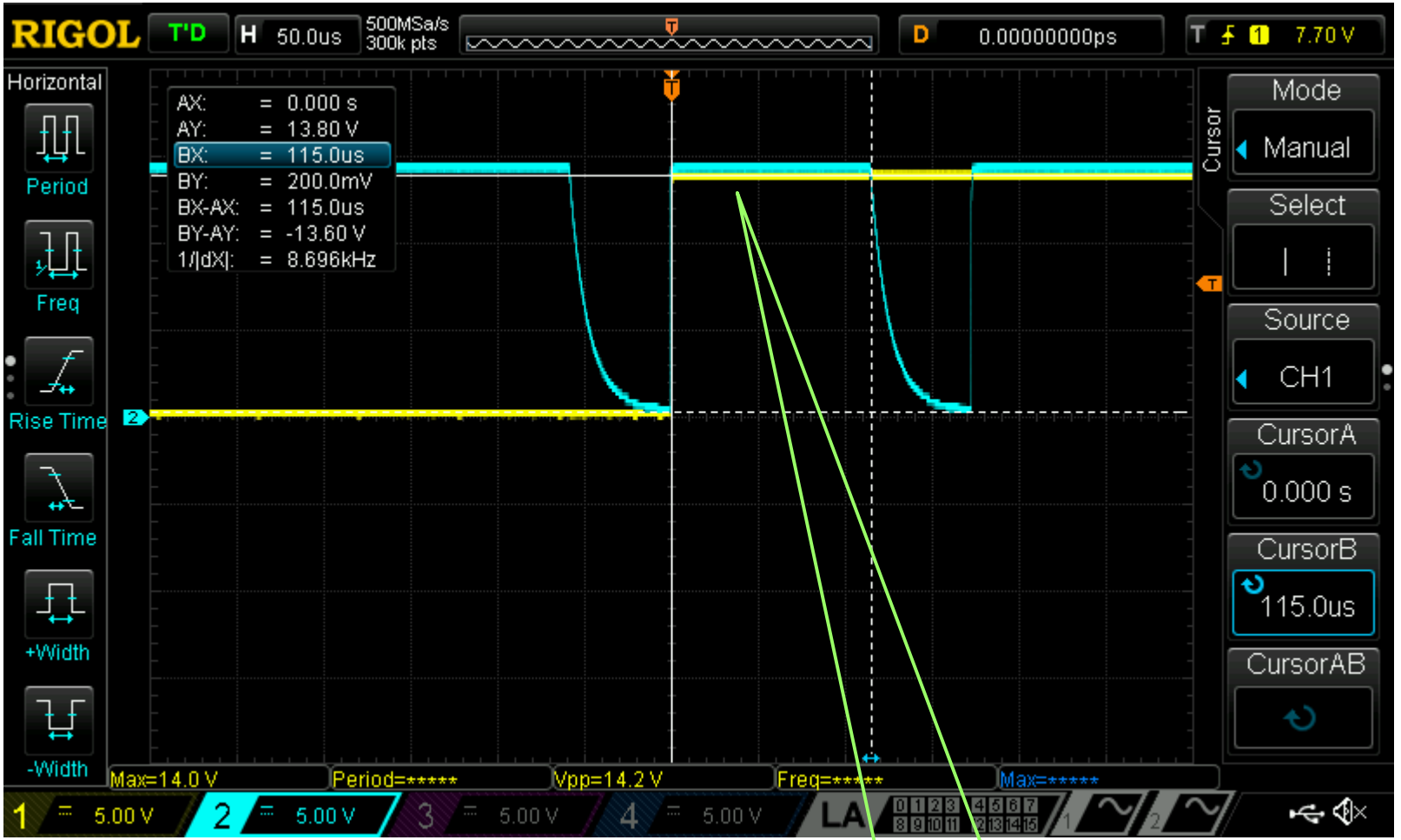
Bit 0, first time slot active low

Display 000000000001 Zoom



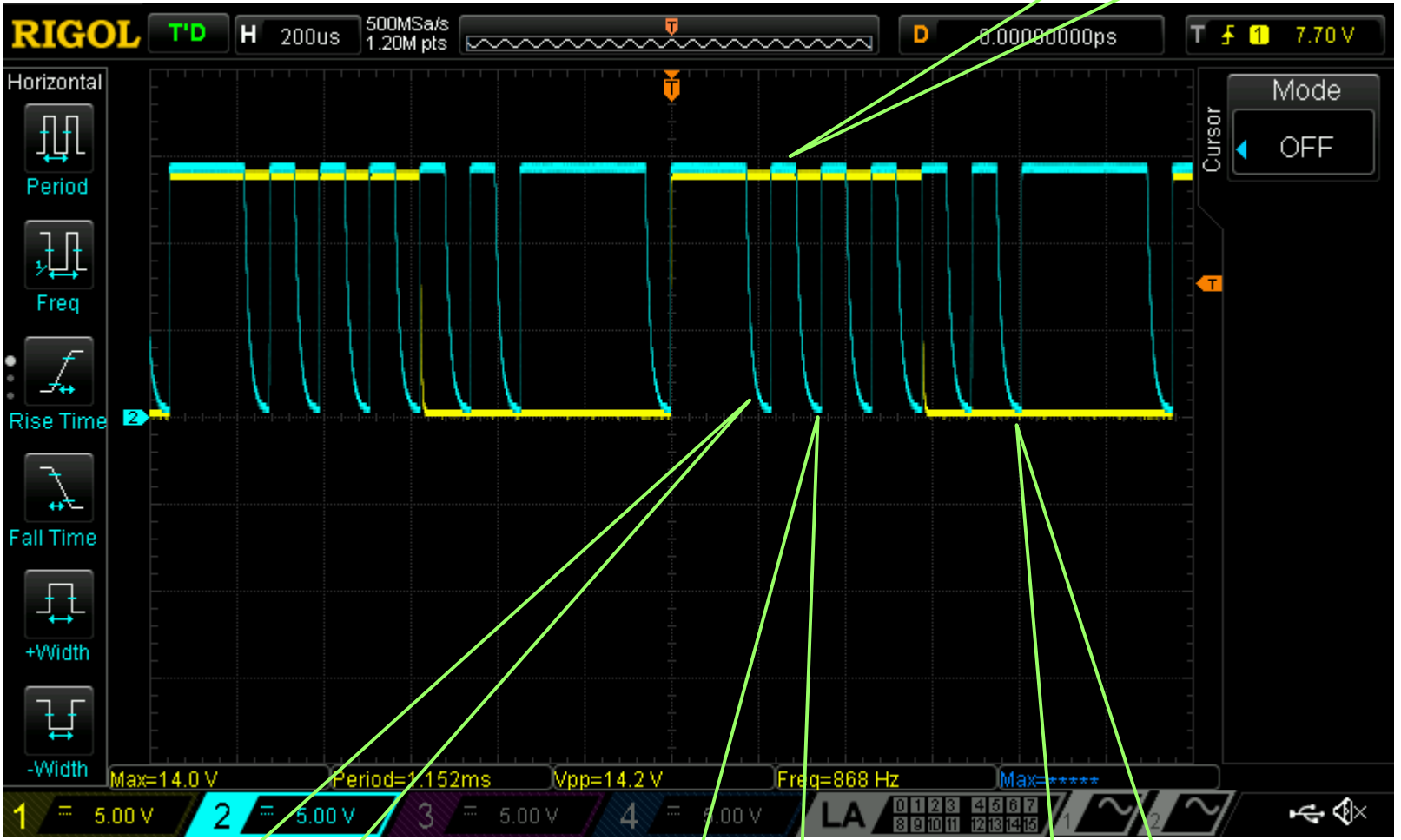
Bit 0, first time slot active low

Display 000000000001 Preamble



115uS high preamble

Display 101010101010



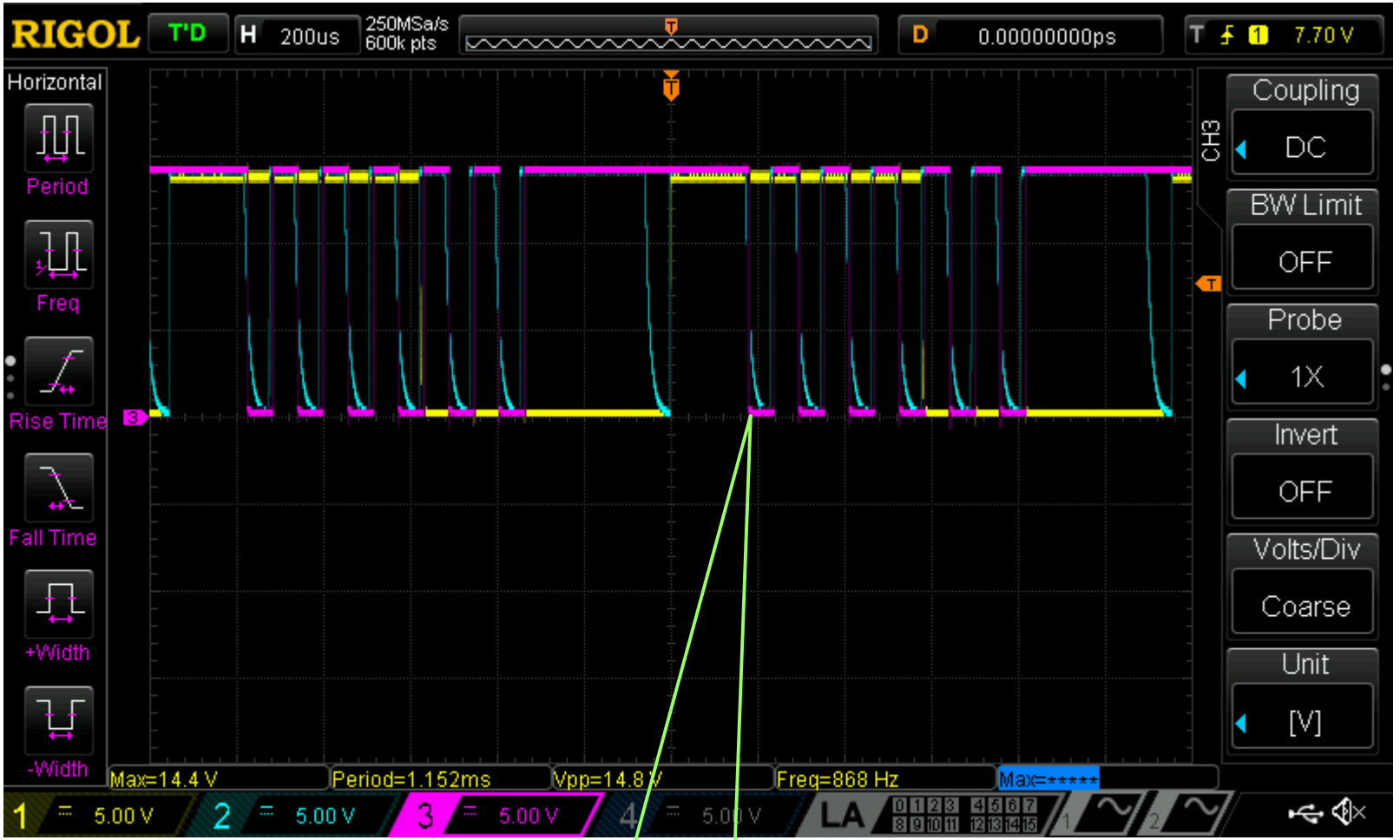
Bit 0, second time slot high (0)

Bit 0, first time slot active low (1)

Bit 0, third time slot active low (1)

Bit 0, last time slot active low (1)

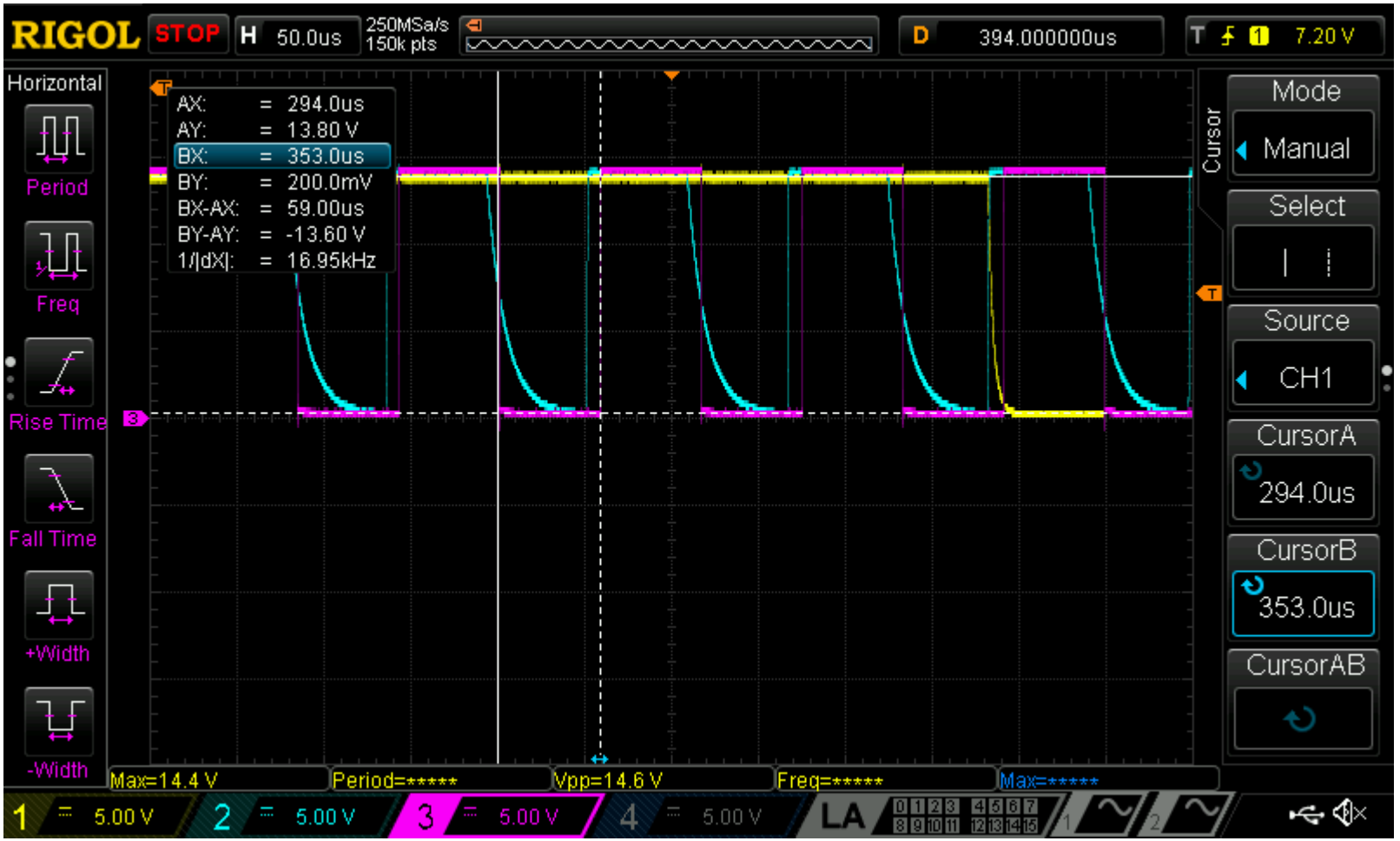
101010101010 Emulation Output



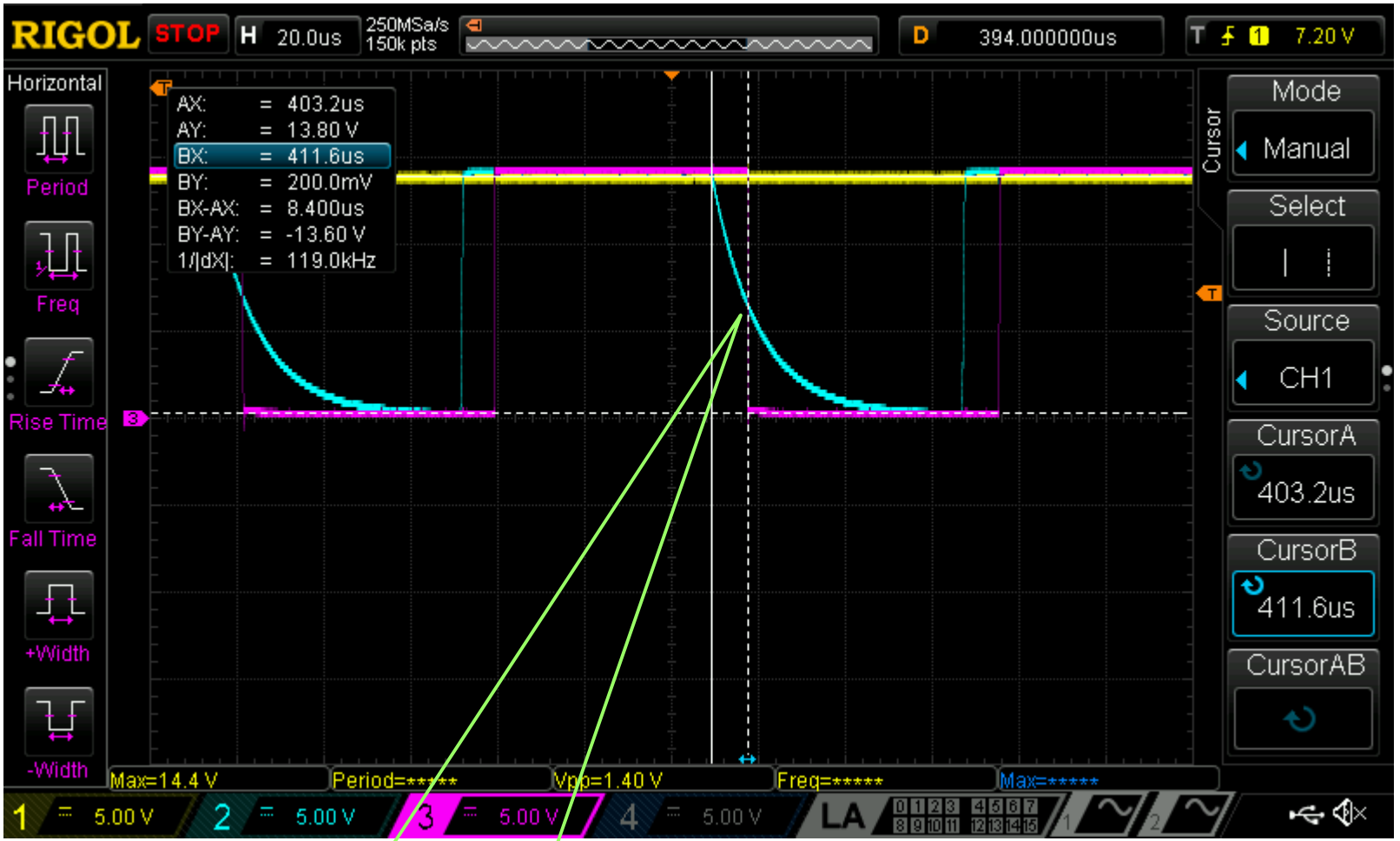
Purple trace is emulated bit

101010101010 Emulation Zoom

101010101010 Emulation Bit Time

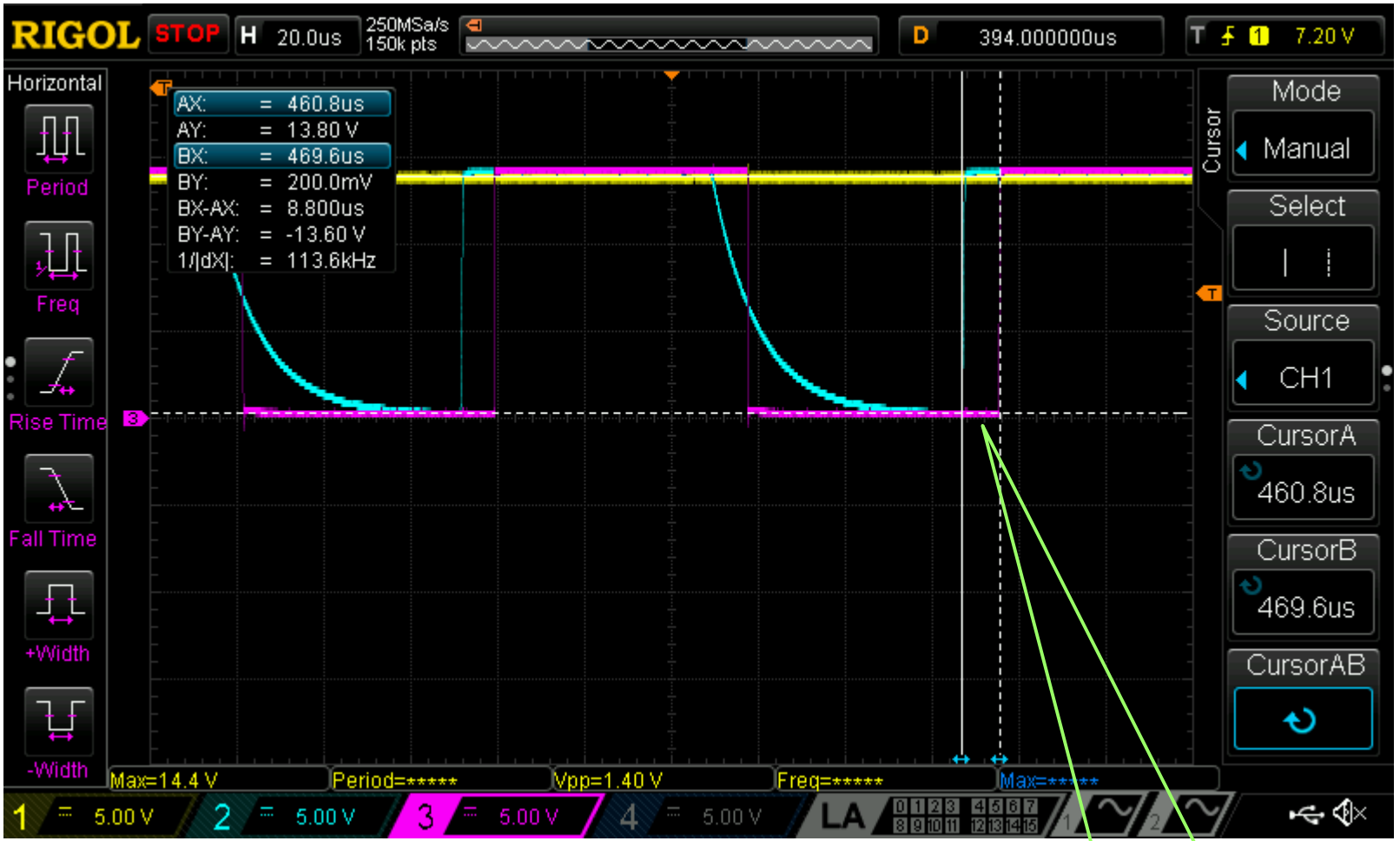


101010101010 Emulation Falltime



Emulation shows much faster falltime so approximate middle of switching threshold

101010101010 Emulation Error



AKA: Don't let perfection be the enemy of good enough

Low for 8.8uS too long;
Fix by varying duty cycle

Future Work

- Run under RTOS (Xenomai, PremptRT, etc)
- Figure out how decimal point works
  - Use as column seperator
- Find bittime clock or do active clock recovery
- Pass-through mode for original calculator logic
- Use as media center
- Support other vintage calculators

Credits

- Nixie images from wikipedia under CC license
  - https://en.wikipedia.org/wiki/Nixie_tube
- Nixie diagram from Radio Electronics (RIP)
  - http://www.decodesystems.com/re-how-nixies-work.html
- App icon CC license from Virginio Savani
  - https://commons.wikimedia.org/wiki/File:Nixie_IN17.svg
- EC1117 background from Rick Bensene
  - http://www.oldcalculatormuseum.com/friden1117.html