uC102: Embedded Communication Protocols / ADC

Josh Johnson

8/7/2019

Overview

- Embedded communication requirements
- UART - Universal Asynchronous Receiver Transmitter
- Tangent: Debugging
- SPI - Serial Peripheral Interface
- Tangent: Manchester Encoding
- I2C - Inter-Integrated Circuit
- JTAG - Joint Test Action Group
- ADC - Analog to Digital Converter

Project Files: `github.com/joshajohnson/CBRhardware`

Embedded communication requirements

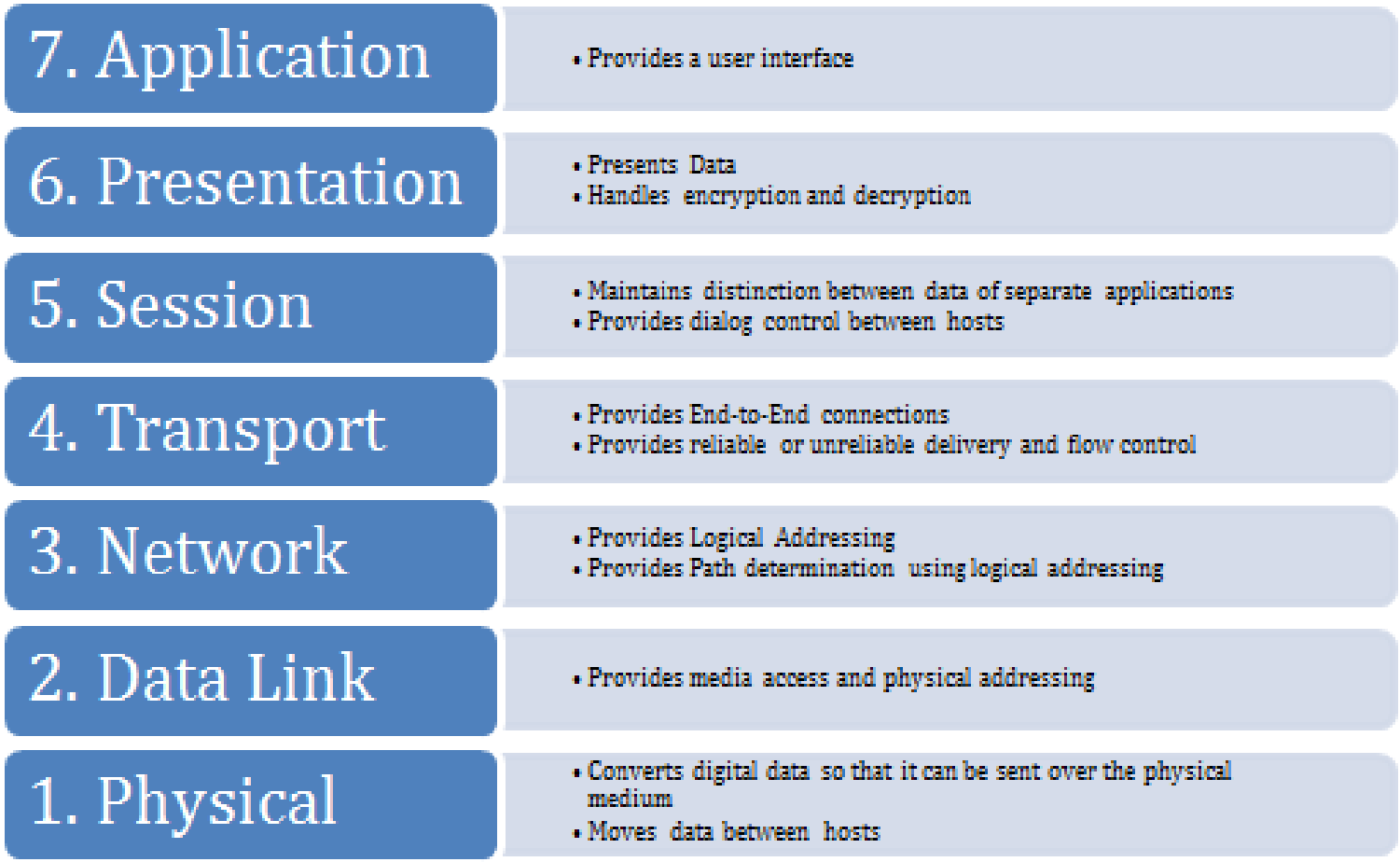
- Typically an embedded processor communicating with peripherals.
- Typically single master, multiple slave system.
- Multi master (eg CAN) does exist, out of scope for today.
- UART/SPI/I2C is probably over 99% of embedded communications.

Common peripherals

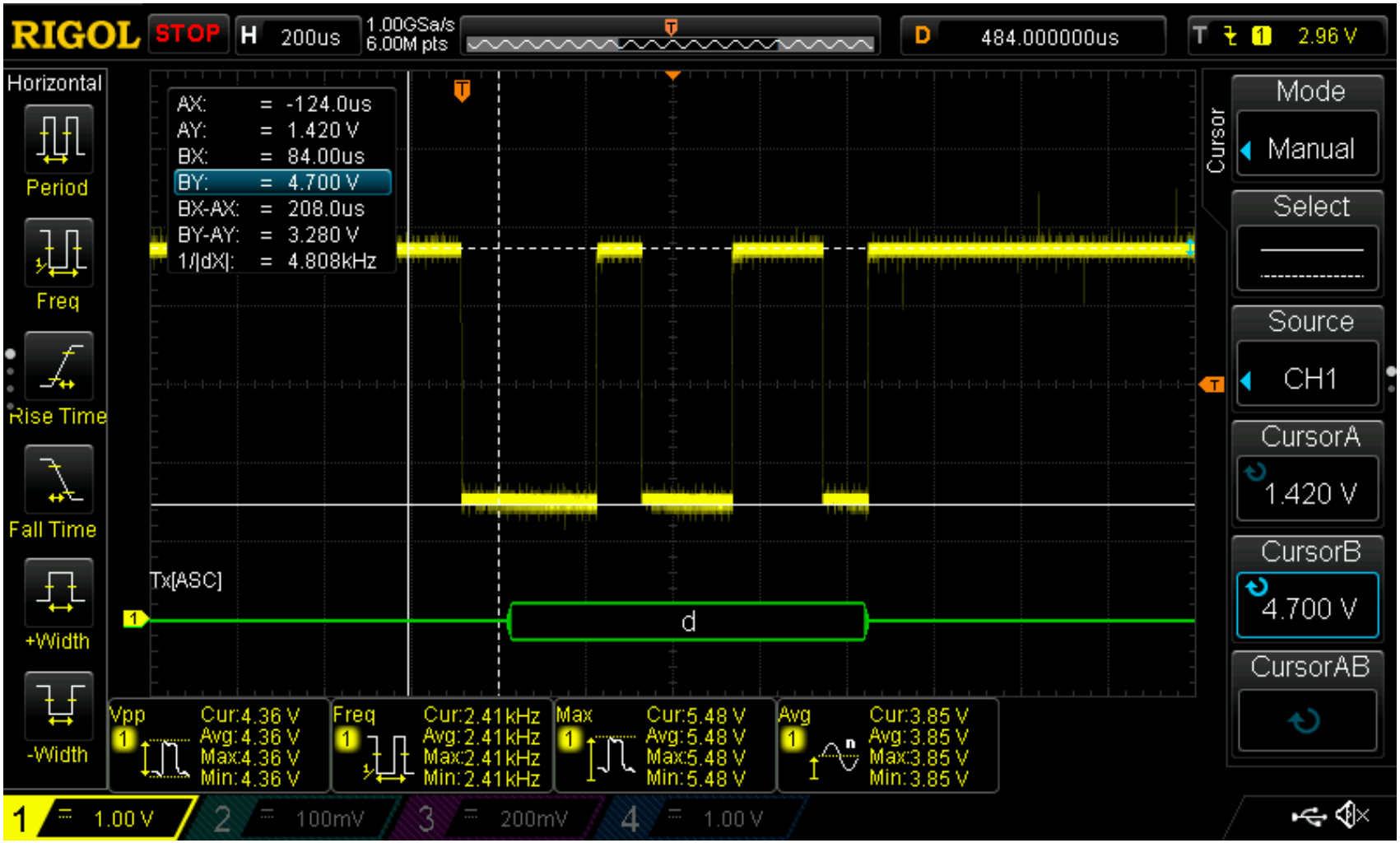
- Inertial Measurement Unit (Gyroscope, Accelerometer)
- SPI Flash Chip / EEPROM
- RF Transmitter
- Display
- Addressable LEDs

Where on the OSI Model?

We are dealing with layer one and two of the OSI model.



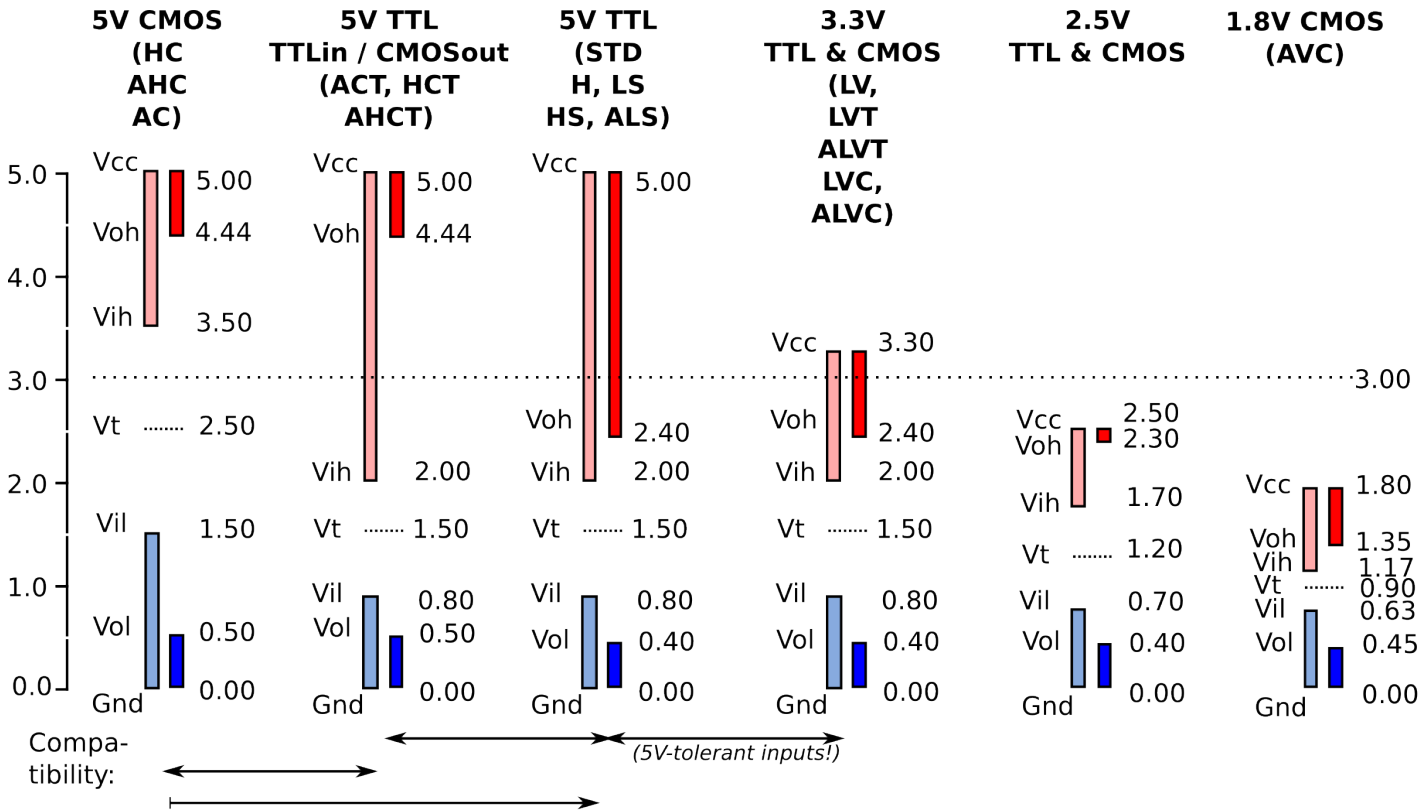
What Physical Medium?



Oscilloscope trace of a (bad) 5V CMOS signal.

What Physical Medium?

Digital data over wires, looking at voltage with respect to time.
Dark colour is output voltage from chip, light input voltage to chip.

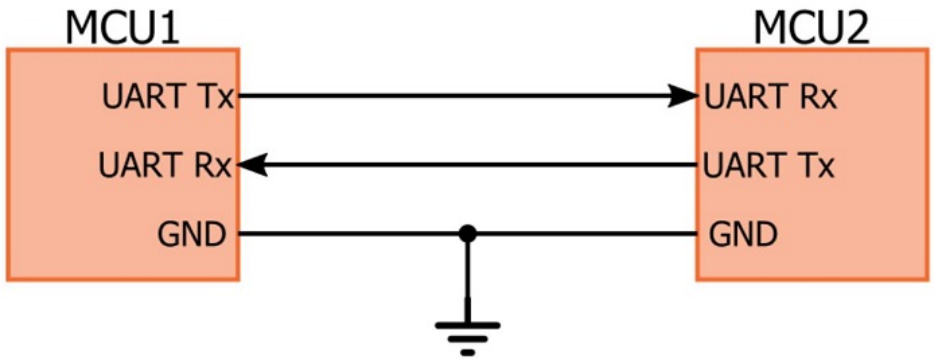


Data source: EETimes, A brief recap of popular logic standards (Mark Pearson, Maxim).

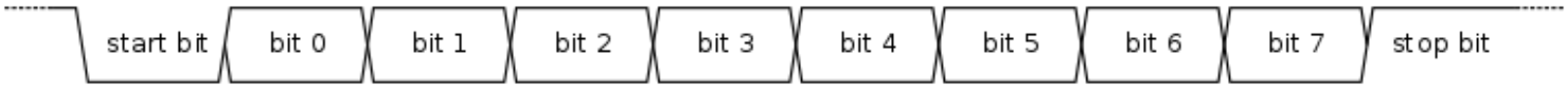
Universal Asynchronous Receiver Transmitter

- **Asynchronous:** not coordinated in time, i.e. no shared clock.
- Two wires, TX and RX.
- Device 1 TX connects to Device 2 RX, and vice versa.
- Accurate clock required for devices to stay in sync.
- Both devices must agree on frame for successful communication.
  - Bits per second (baud rate).
  - Bits per transfer.
  - Number of stop bits.
  - Parity bit (even/odd/none).
  - LSB / MSB first.
  - Inverted.
- 9600 baud, 8N1 is the most common (8 bits per transfer, no parity, 1 stop bit).

UART



Typical UART connection.

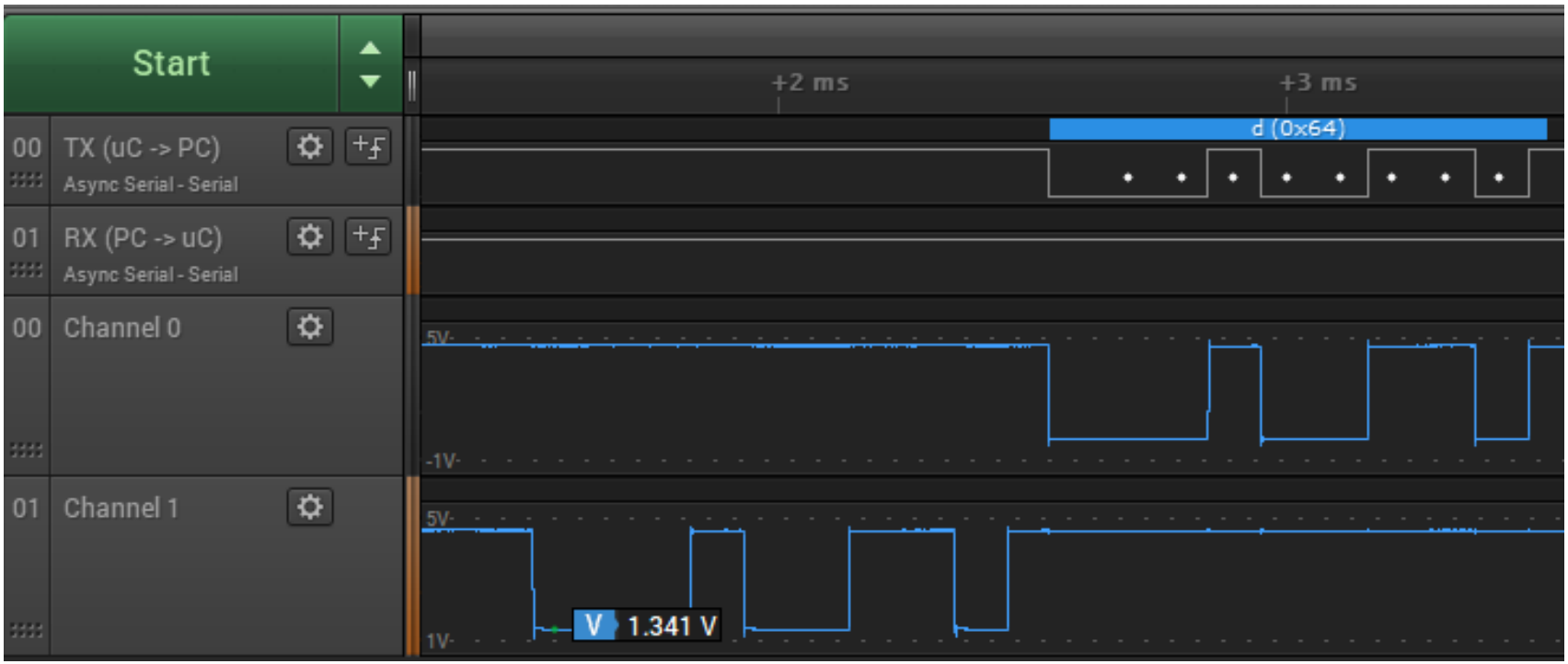


Typical UART data frame.

Regarding the need for accurate clocks:
Ben Eater: https://youtu.be/eq5YpKHXJDM?t=1590 (26:30 in)

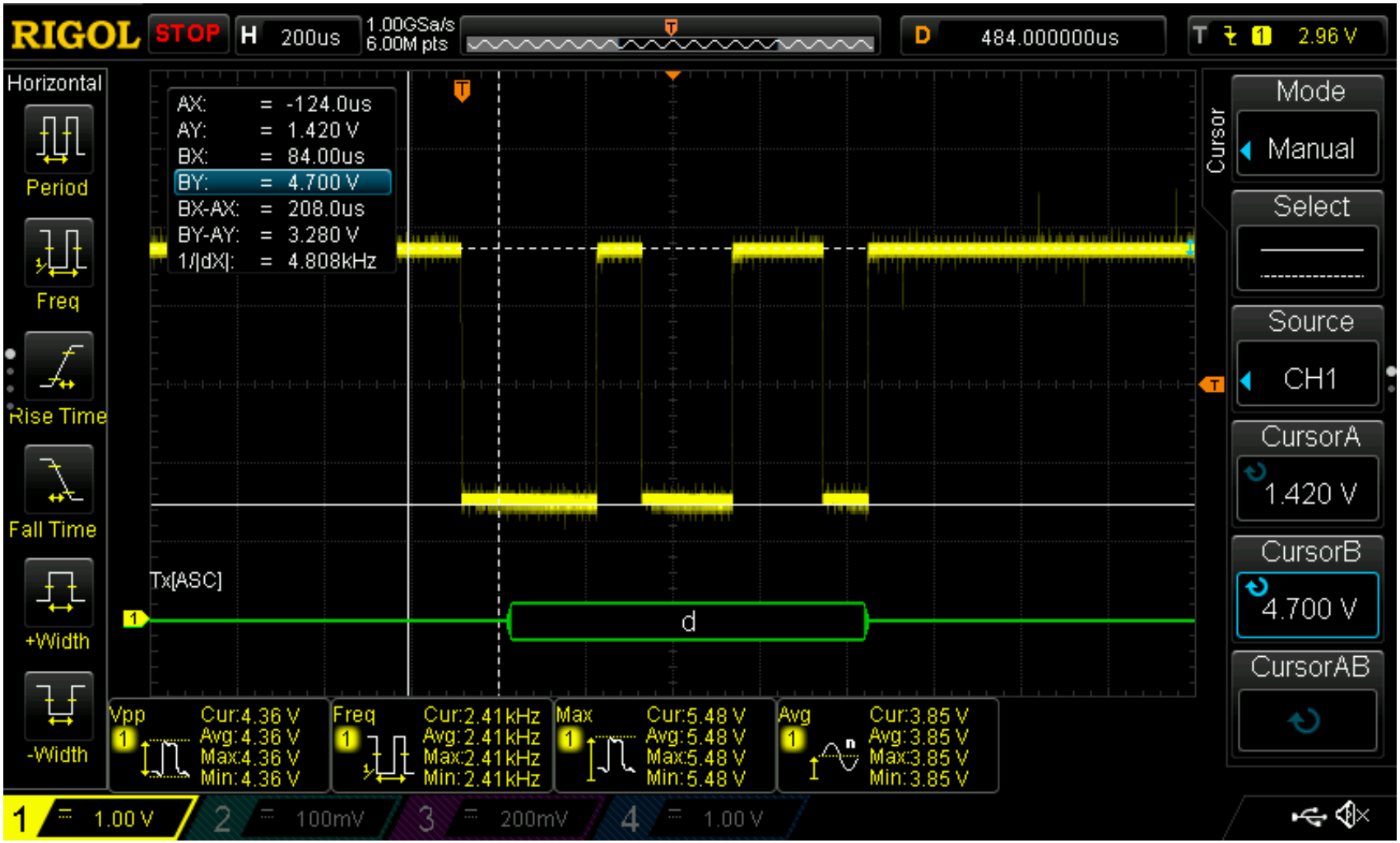
How to Debug

- Oscilloscope
- Logic Analyser
- Try known good HW / SW



Logic capture of the UART comms with my PC.

How to Debug

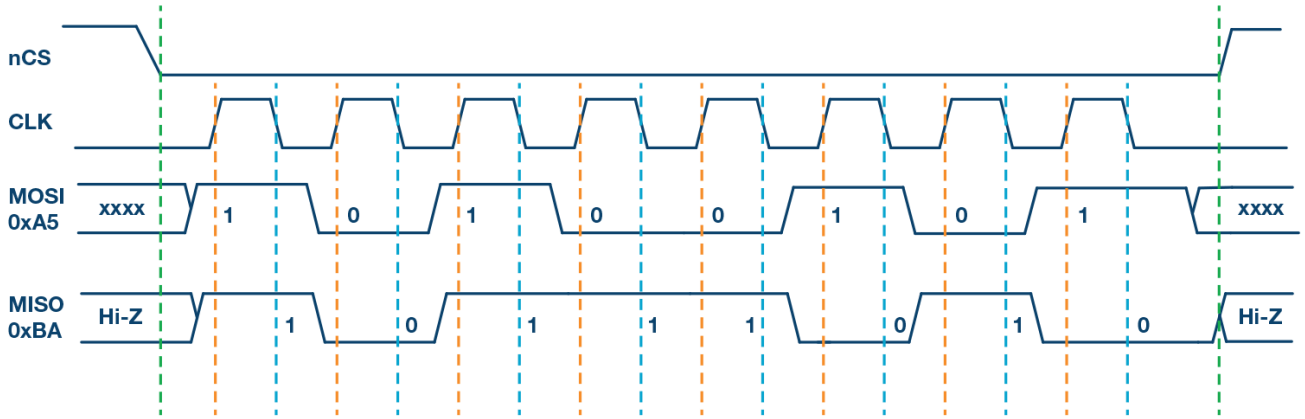


Note the low voltage of 1.4V, not the 0 we expected.

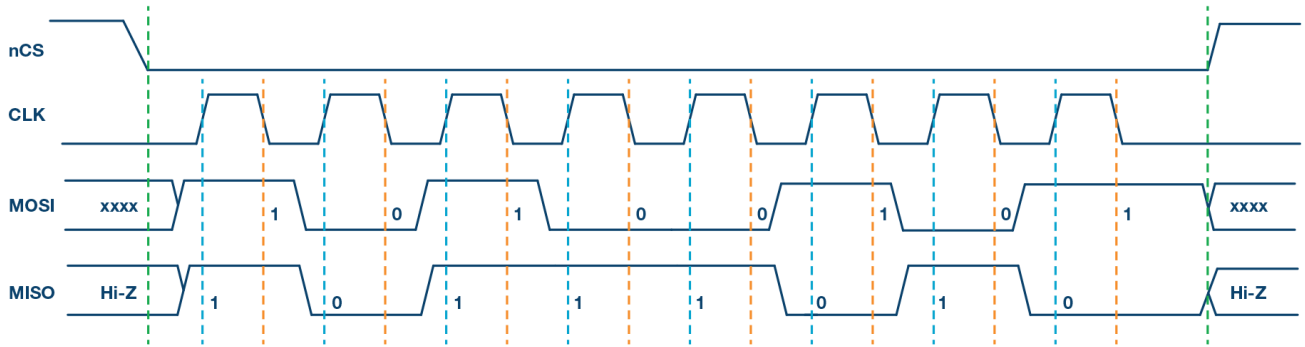
Serial Peripheral Interface

- Synchronous, full duplex master-slave-based interface.
- 4 wires, SCLK, MOSI, MISO, nCS.
- Data is synchronised on rising or falling clock edge.
- Single master, data simultaneously transmitted in and out.
  - Clock Polarity (CPOL)
  - Clock Phase (CPHA)
  - MSB vs LSB first
- To determine, check the timing diagrams for your specific chip, as we will see tonight SPI has many implementations.

SPI



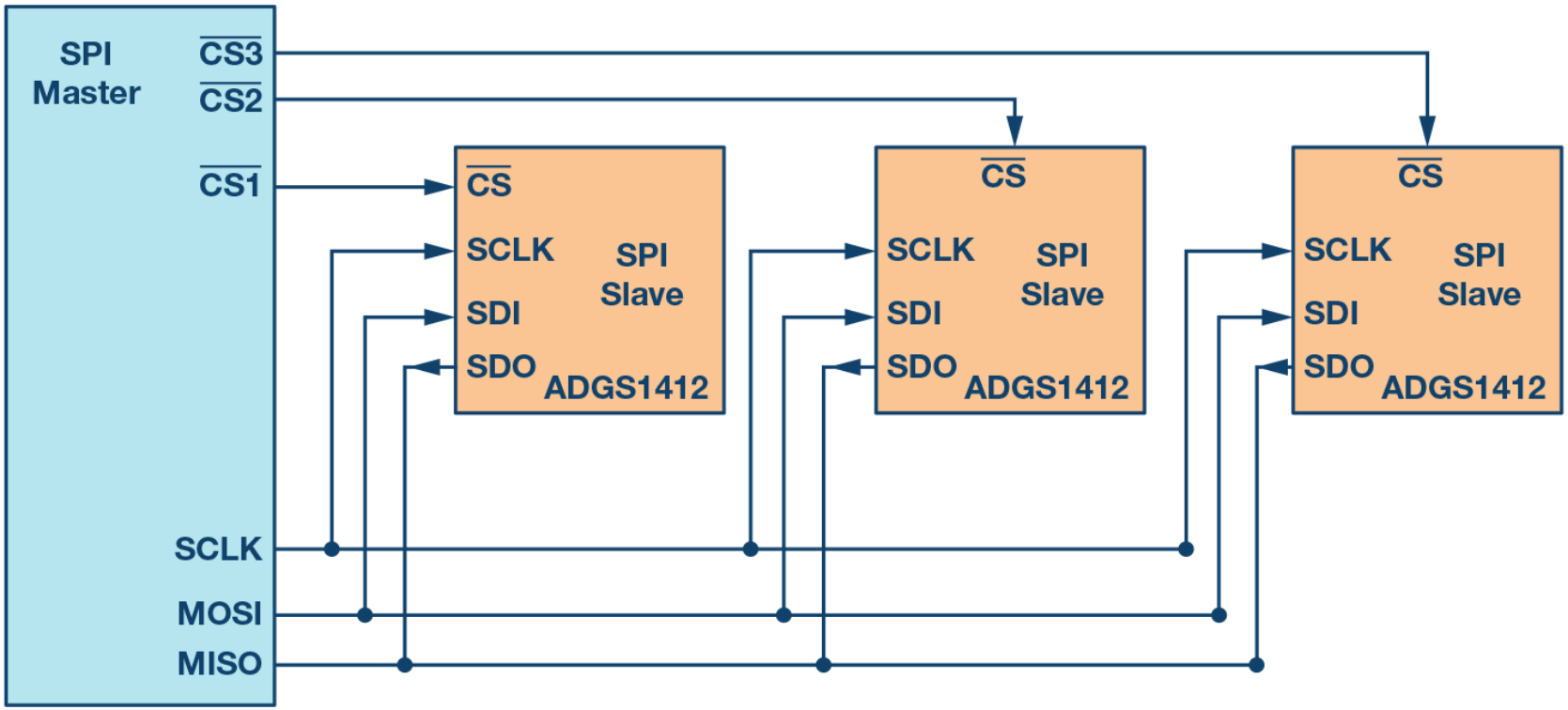
SPI Mode 0, CPOL = 0, CPHA = 0: CLK idle state = low, data sampled on rising edge and shifted on falling edge.



SPI Mode 1, CPOL = 0, CPHA = 1: CLK idle state = low, data sampled on the falling edge and shifted on the rising edge.

SPI

Multiple devices can be connected to same SPI bus if they use the same communication settings.



Regular multislave SPI configuration.

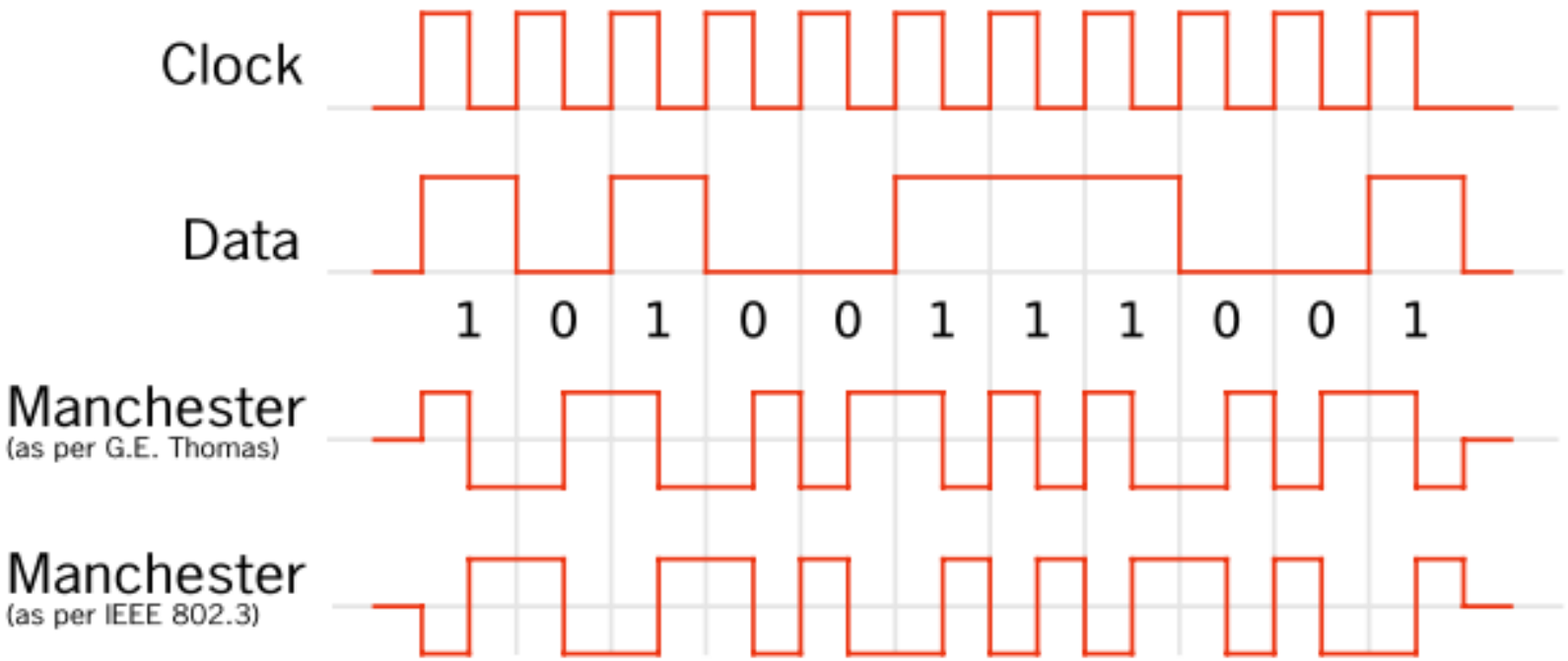
SPI

Datasheet Time!

- APA102 addressable LEDs
- MAX2871 PLL

Tangent: Manchester Encoding

Manchester encoding integrates a clock into the transmitted signal, however halves the bandwidth.



Used in car keyfobs, 10Base-T ethernet (IEEE 802.3), amongst others.

Tangent: Manchester Encoding

Encoding data using exclusive or logic (802.3 convention)[4]

Original data		Clock		Manchester value
0	XOR ⊕	0	=	0
		1		1
1		0		1
		1		0

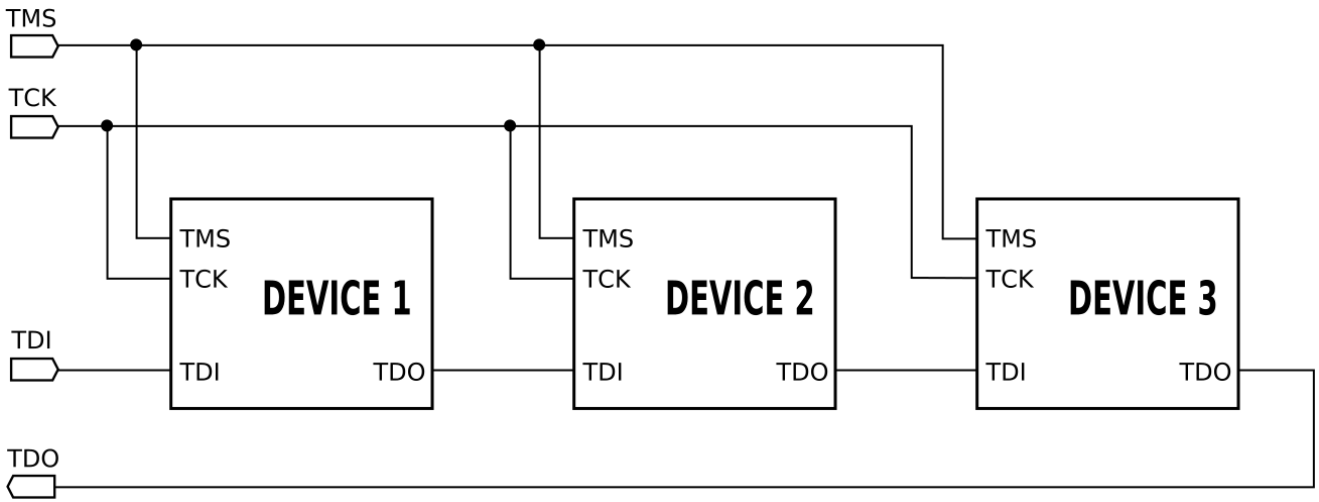
Calculating the Manchester encoded signal.

```
for (uint8_t i = 0; i < numberOfBytes; i++)
{
    if (data[i])
    {
        while(TCNT1 < HALF_PERIOD_ONE);
        digitalWrite(TX_PIN, 1);

        while(TCNT1 < TIME_PERIOD);
        digitalWrite(TX_PIN, 0);

        TCNT1 = 0;
    }
    else
    {
        while(TCNT1 < HALF_PERIOD_ZERO);
        digitalWrite(TX_PIN, 0);

        while(TCNT1 < TIME_PERIOD);
        digitalWrite(TX_PIN, 1);

        TCNT1 = 0;
    }
}

while (TCNT1 > TIME_PERIOD/2);
digitalWrite(TX_PIN, 0);
```

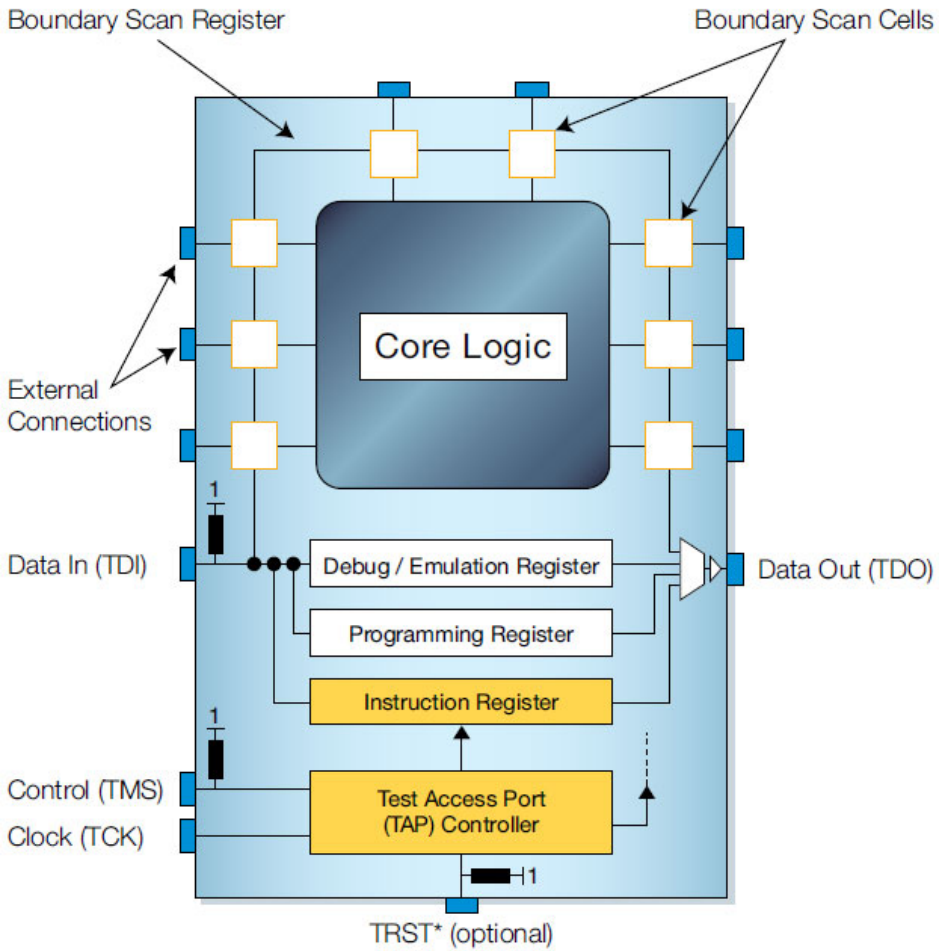
Arduino implementation of Manchester encoding

Joint Test Action Group

- 4 wires, TCK, TDI, TDO, TMS.
- Serial connection similar to SPI, however devices are daisy chained.
- Originally designed to test complicated (BGA, high density) boards.
- Later used to program and debug ICs.
- SWD uses a reduced pin count version of JTAG with two wires (SWDIO, SWCLK).



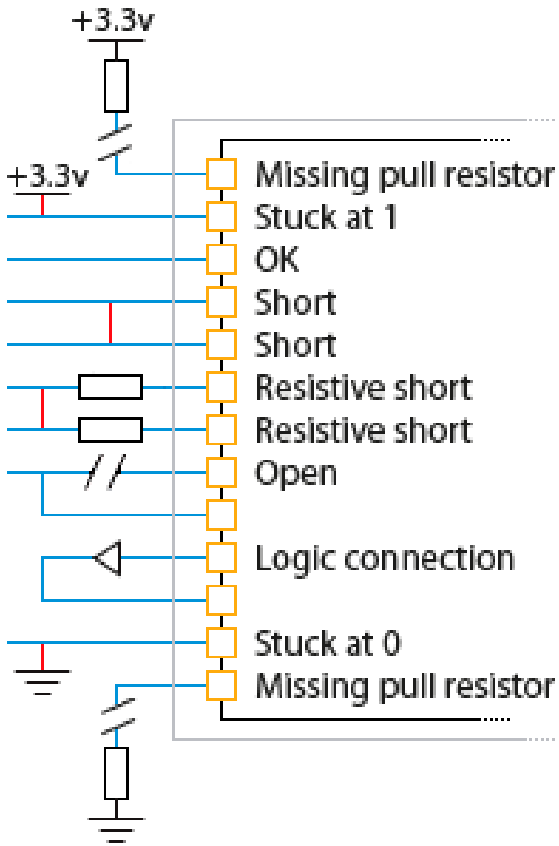
JTAG



Internal architecture of an IC supporting JTAG.

Boundary Scan

- Automated testing of all connections on board.
- Utilises BSDL (Boundary Scan Description Language) file for each component, which defines pin types.
- Can test components on single board, or connect multiple boards together to scan complicated system.
- Looks for stuck high, stuck low, pins tied together, not connected.
- Requires external software and programmer to configure.
- Used in production testing to check for bad connections.



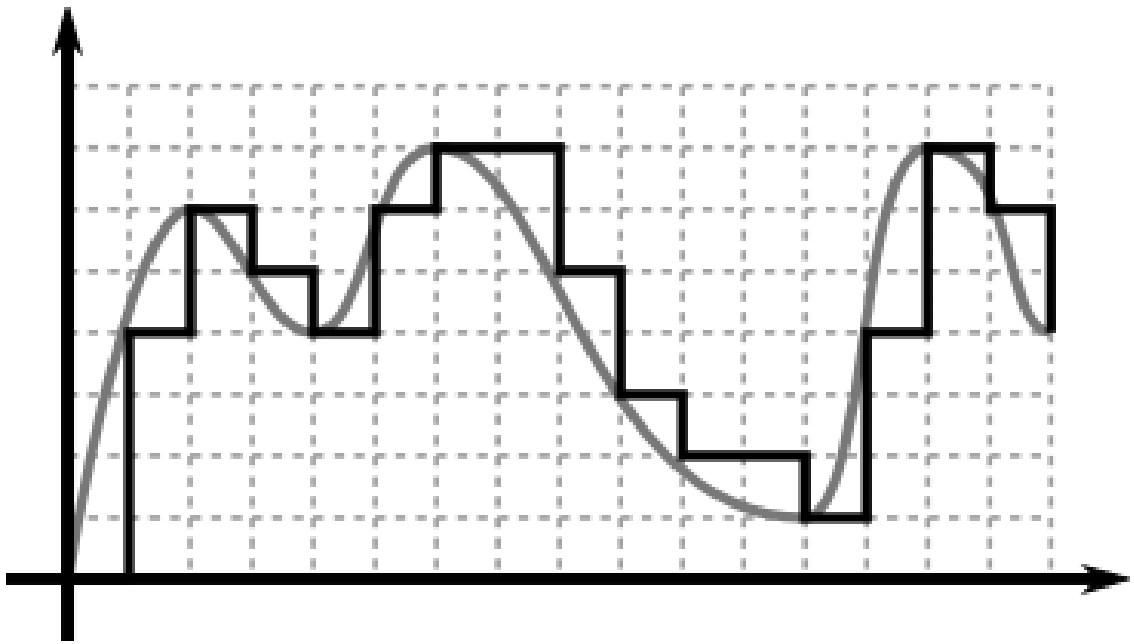
JTAG

Programming / Debugging

- Can be used to program flash / configure FPGA / etc.
- Low level connections into IC allow for register level debug.
- Manufactures implement their own commands which can be called by JTAG to set breakpoints, read registers etc.
- Due to use in production, JTAG often left exposed on PCB with no protections.
- This allows user to control JTAG device if they have physical access.
- Devices (e.g. ATmega) may have JTAG, but may not be used. ICSP (In Circuit Serial Programming) utilised instead to program device.

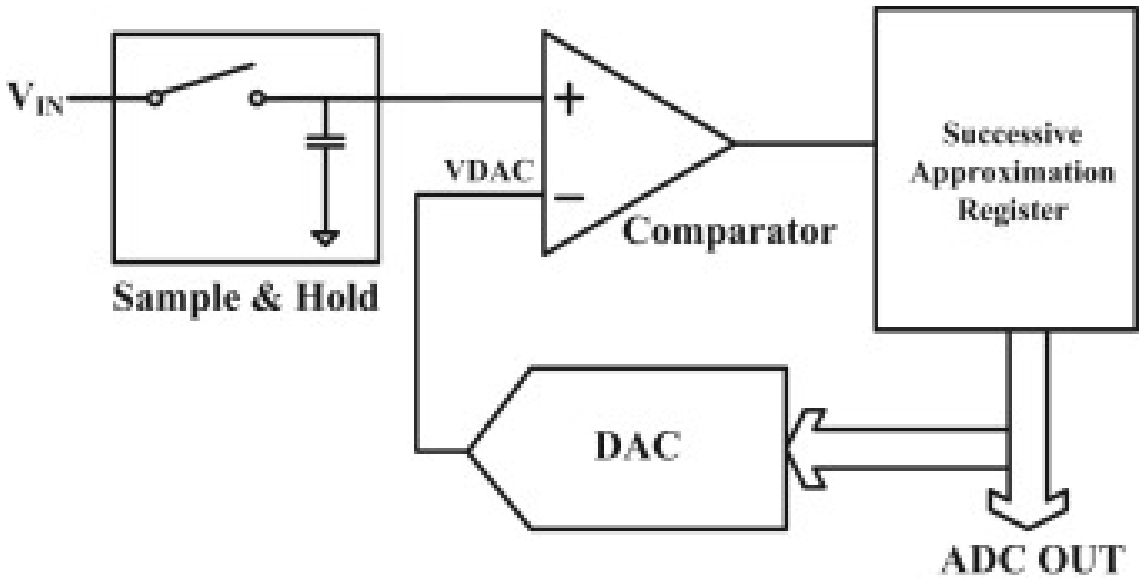
Analog to Digital Converter

- Converts an analogue signal to a digital representation.
- More bits = better representation of signal.
  - E.g. 10 bits at VREF = 5V = 4.8mV resolution.
  - E.g. 16 bits at VREF = 3V3 = 50μV resolution.
- Conversion takes time, on AVR it is 13 clock cycles.



ADC

- Two most common: successive approximation register and sigma-delta.
  - SAR converters are faster, but lower resolution. (1 MSPS, 12-16 bits).
  - S-D are slower, but much higher resolution (100 kSPS, 16-24 bits).
- Need to ensure output impedance to ADC is low enough, otherwise not enough current will flow into S/H capacitor.
- If sampling time varying signals, ensure no signal components higher than Nyquist.



Voltage to ADC bits

$$\text{ADC Bits} = \frac{V_{\text{IN}} \cdot 2^N}{V_{\text{REF}}}$$

Voltage from ADC bits

$$\text{Voltage} = \frac{\text{ADC Bits} \cdot V_{\text{REF}}}{2^N}$$

Where N is the number of bits.

The End

Links to resources: `uC102/README.md`

Next month

- Further PCB Design / Manufacturing / KiCad
- Intro to FPGA
- Someone else's talk!

Say Hello!
BSidesCbr Slack: josh
Twitter: @_joshajohnson
Email: josh@joshajohnson.com

Project Files: `github.com/joshajohnson/CBRhardware`