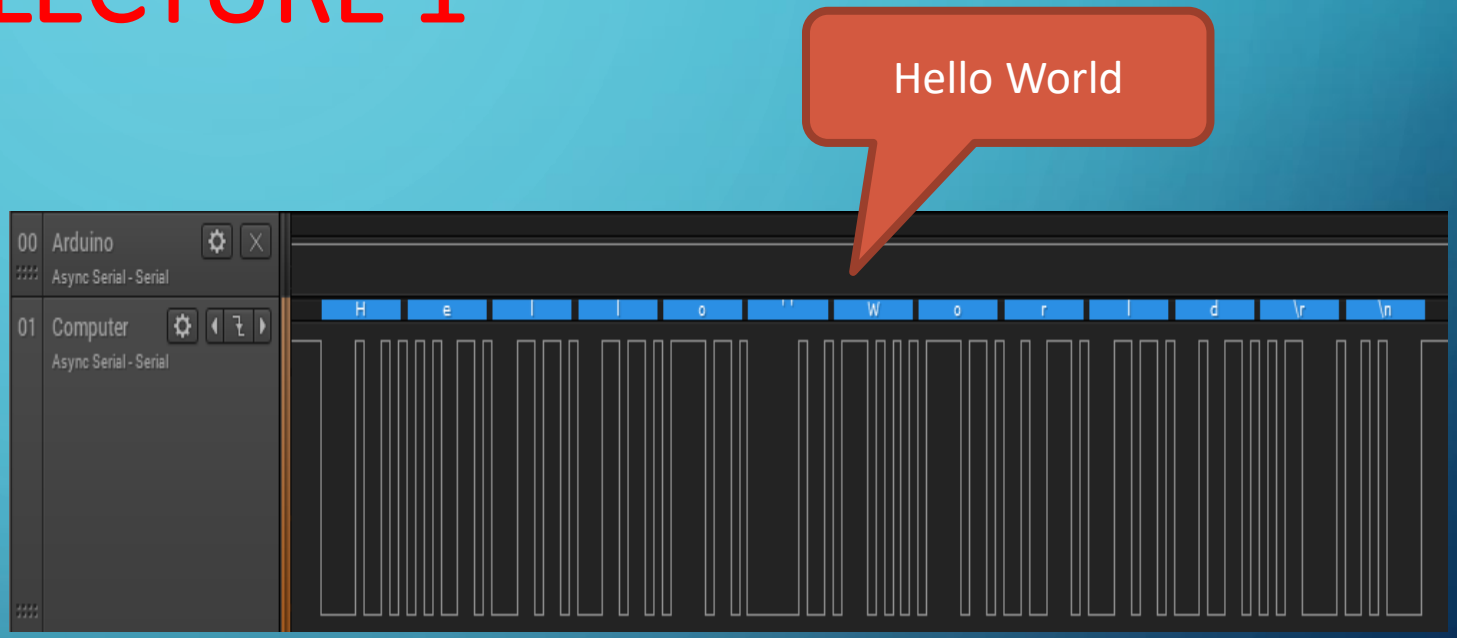
COMMUNICATION PROTOCOLS REVERSE ENGINEERING

LECTURE 1



BY: OMAR MEKKAWY

MY LINKEDIN: HTTPS://WWW.LINKEDIN.COM/IN/OMAR-MEKKAWY/

GITHUB: HTTPS://GITHUB.COM/RXTXINV/COMMUNICATION_PROTOCOLS_REVERSE_ENGINEERING_COURSE/

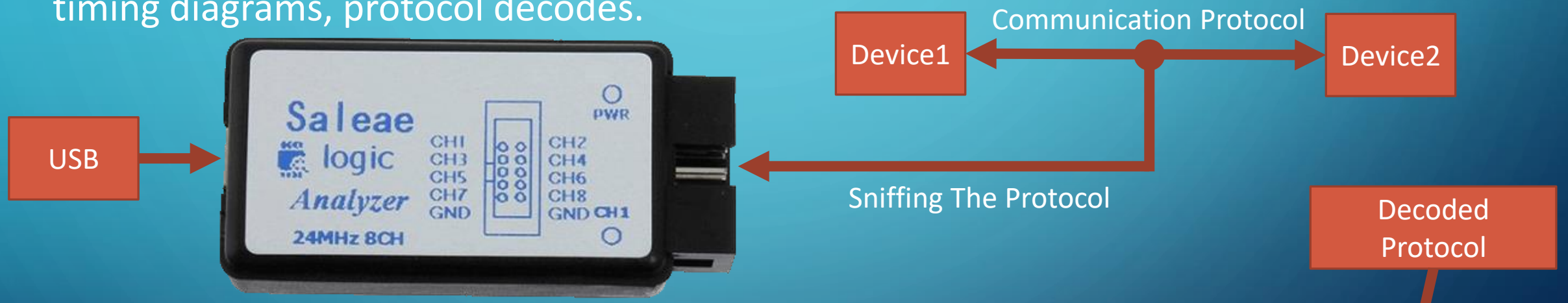
MY WEBSITE: HTTPS://OMAR-MEKKAWY.NET – HTTPS://OMAR-MEKKAWY.COM

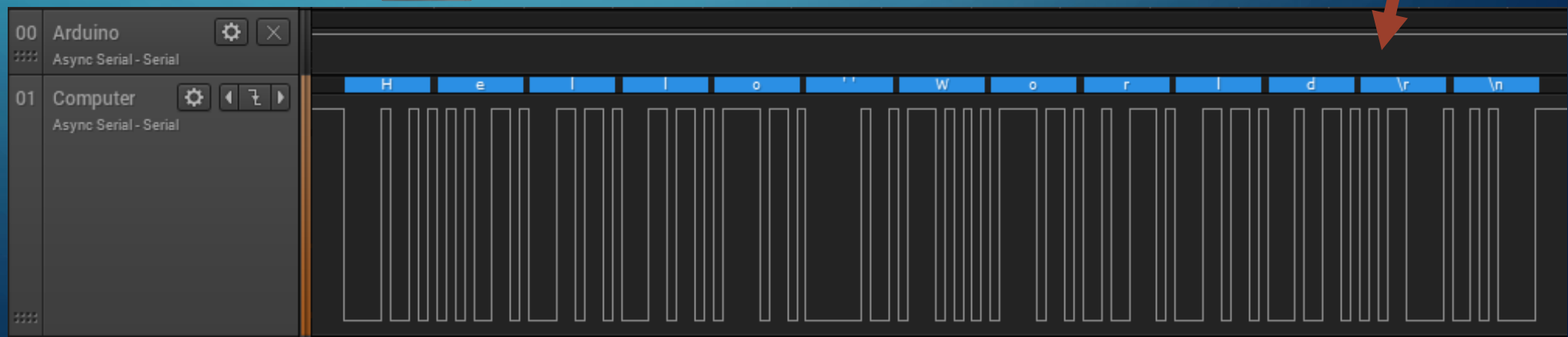
LECTURE 1 AGENDA

- What is the Logic Analyzer ?
- What is the importance of the logic analyzer ?
- The physical components of the logic analyzer.
- The Software interface.
- Why using Arduino in this course ?
- Example 1 (LED Blink)
- Example for Sniffing UART Protocol.

WHAT IS THE LOGIC ANALYZER

- The logic analyzer is an electronic instrument that captures (Samples) and displays multiple signals from a digital system or digital circuit **without affecting the communication between them**. A logic analyzer may convert the captured data into timing diagrams, protocol decodes.

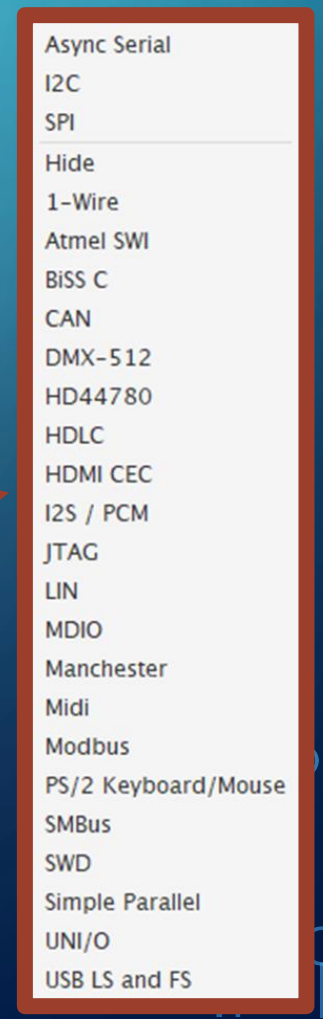




WHAT IS THE IMPORTANCE OF THE LOGIC ANALYZER

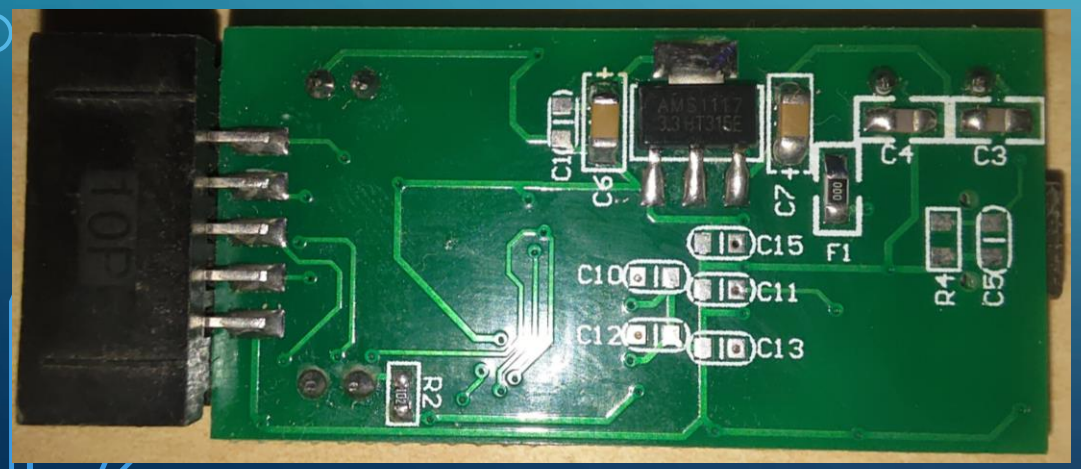
- Its importance:
  - Useful tool when **learning the communication protocols** (UART, I^2C, SPI, 1-Wire, CAN, LIN, SMBus, I^2S).
  - Useful tool when debugging the embedded hardware.
  - Useful tool when trying to reverse-engineer the communication protocols for a product.
  - Capturing the data for later use and documentation purposes.
- Its considered as a **skill** for most of the embedded systems companies.
- Supported Protocols:
  - I^2C, SPI, 1-Wire, CAN, I^2S – PCM, JTAG, LIN, Modbus, SMBus, … more.
- Support 8 Digital Channels with sampling rate up to 24Ms/S.

Supported Protocols



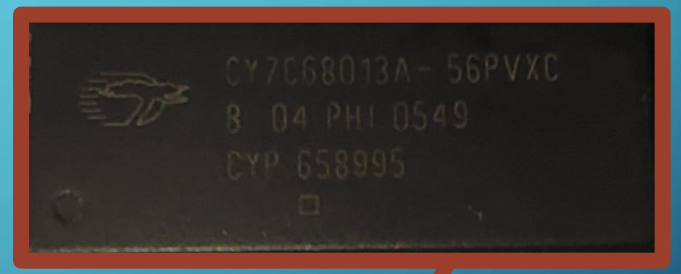
THE PHYSICAL COMPONENTS OF THE LOGIC ANALYZER

- Components of the **Saleae Logic 8** (**Chinese Clone**)
  - Cypress CY7C68013A Microcontroller.
  - 74HC245 (Octal Bus Transceiver) used for **level shifting** and **protection**.
  - 24C02B EEPROM.

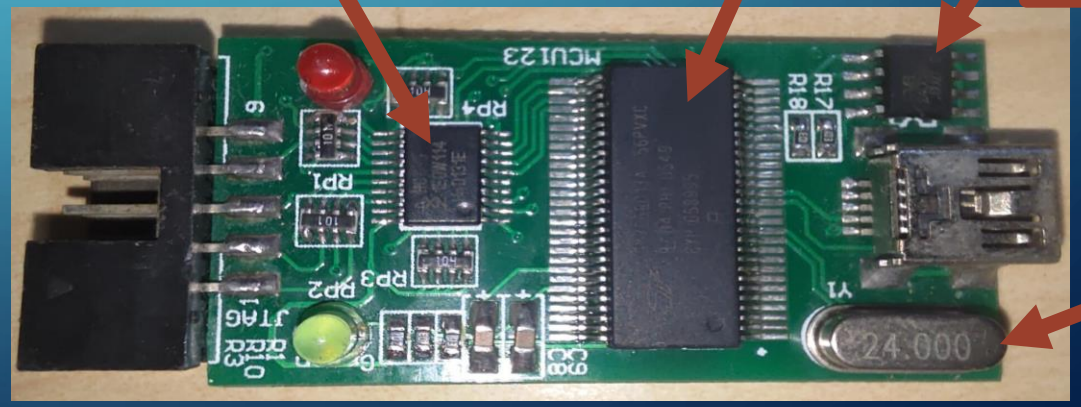


74HC245

CY7C68013A



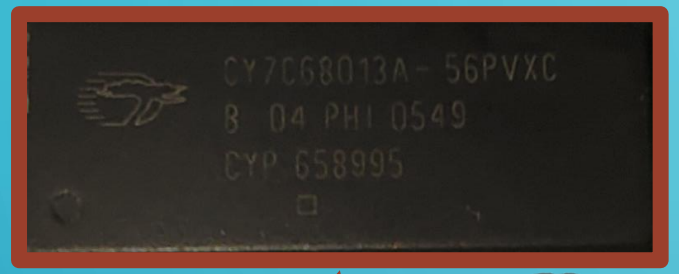
24C02B

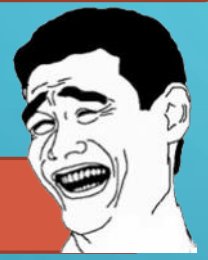


24MHz Crystal

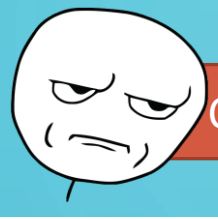
THE PHYSICAL COMPONENTS OF THE LOGIC ANALYZER

Cypress CY7C68013A

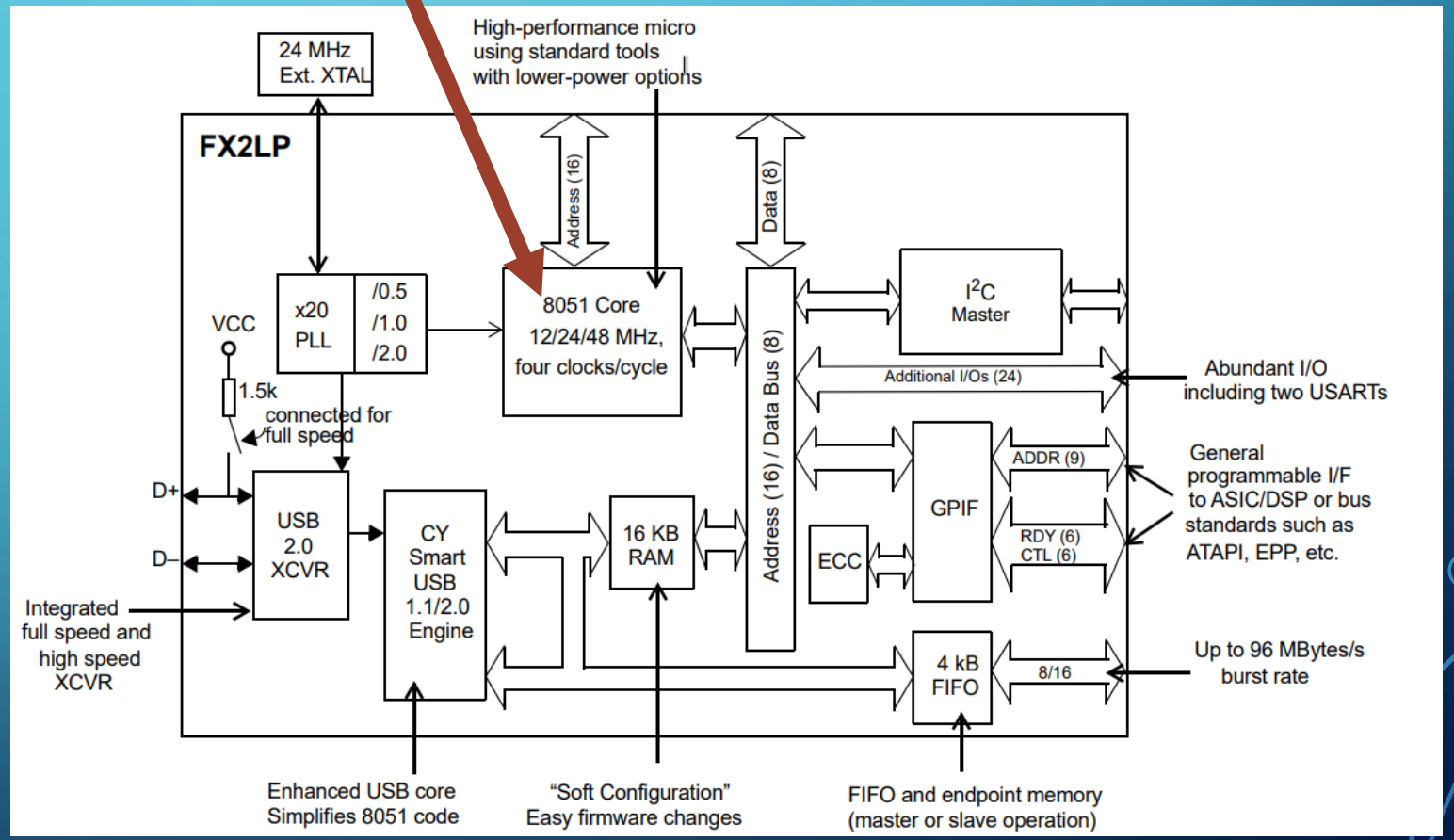




8051 Inside ?!



Confirmed, **8051** Is Here !!



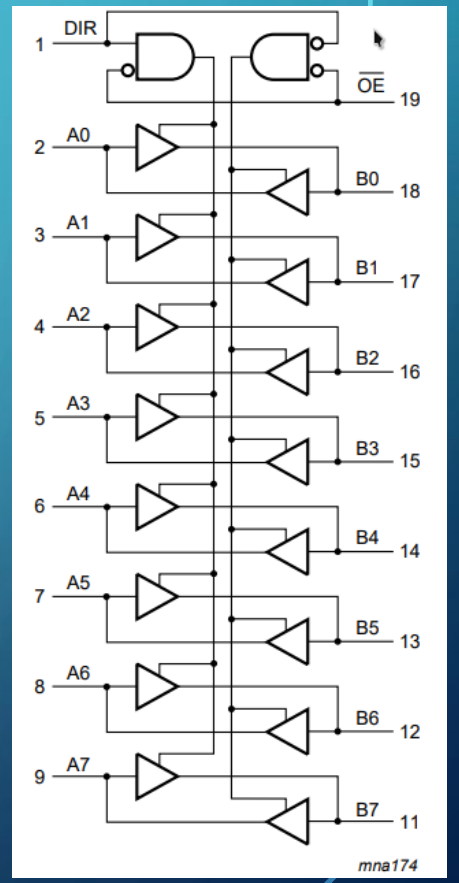
THE PHYSICAL COMPONENTS OF THE LOGIC ANALYZER

74HC245

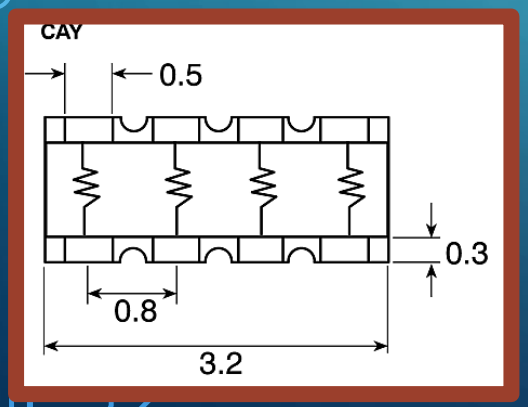
Maximum Input Voltage = 5V

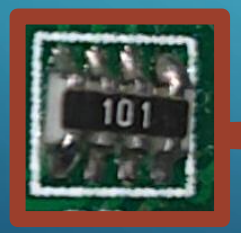
The 74HC245; 74HCT245 is an 8-bit transceiver with 3-state outputs. The device features an output enable ($\overline{OE}$) and send/receive (DIR) for direction control. A HIGH on $\overline{OE}$ causes the outputs to assume a high-impedance OFF-state. Inputs include clamp diodes. This enables the use of current limiting resistors to interface inputs to voltages in excess of V_{CC}.

74HC245

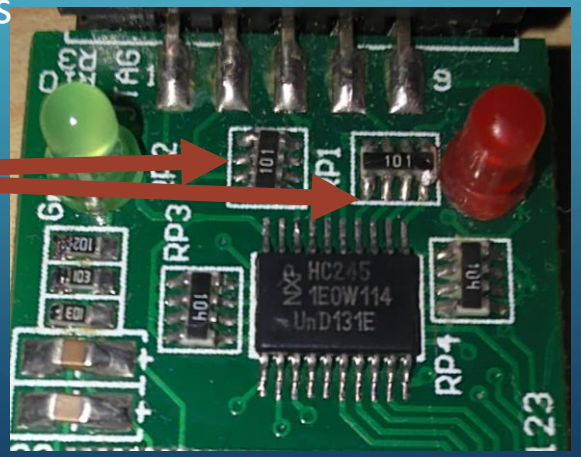


Current Limiting Resistors





100 Ohm Resistor Network

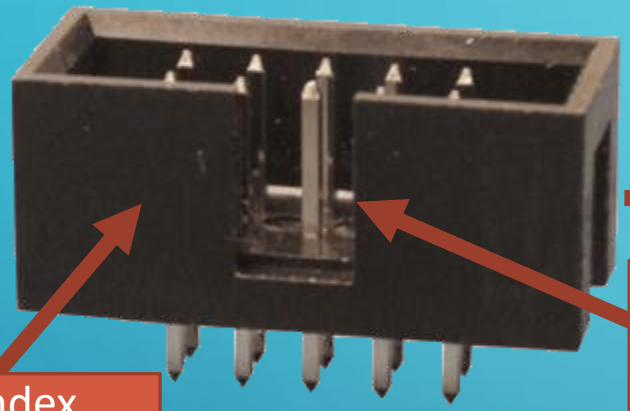


Power LED

Channel 0 LED

THE PHYSICAL COMPONENTS OF THE LOGIC ANALYZER

10Pin IDC Socket



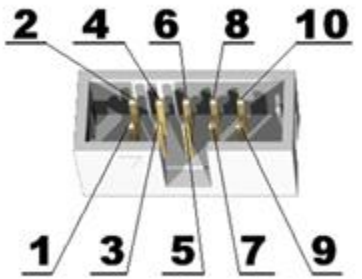
Pin1

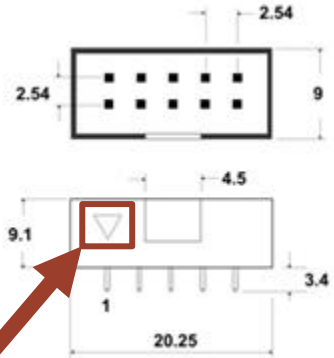
Pinout

Socket Key Slot

Pin1 Index Pointed By Arrow

Channel 0	1	2	Channel 1
Channel 2	3	4	Channel 3
Channel 4	5	6	Channel 5
Channel 6	7	8	Channel 7
GND	9	10	GND

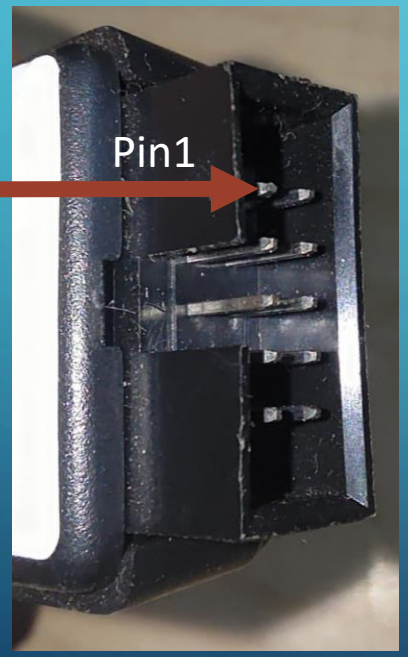


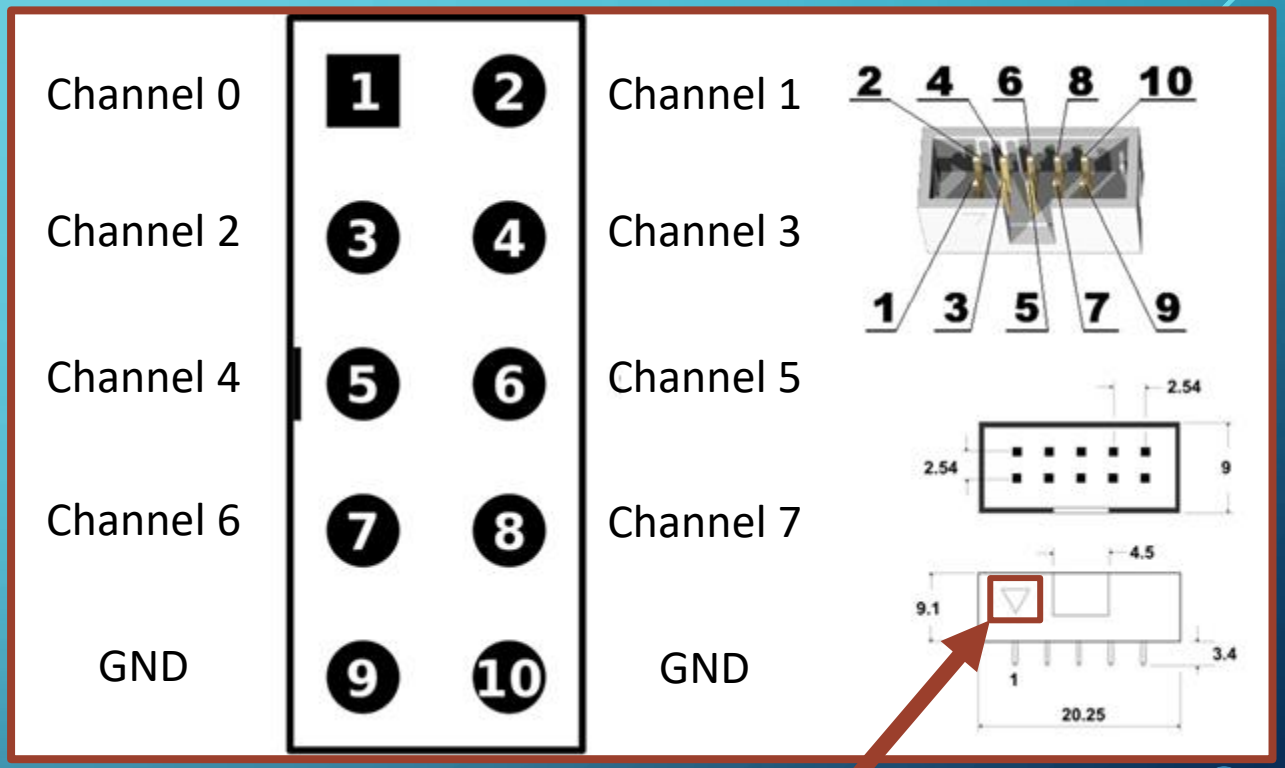


Pin1 Index Pointed By Arrow

THE PHYSICAL COMPONENTS OF THE LOGIC ANALYZER

Pin1 Index Pointed By Arrow

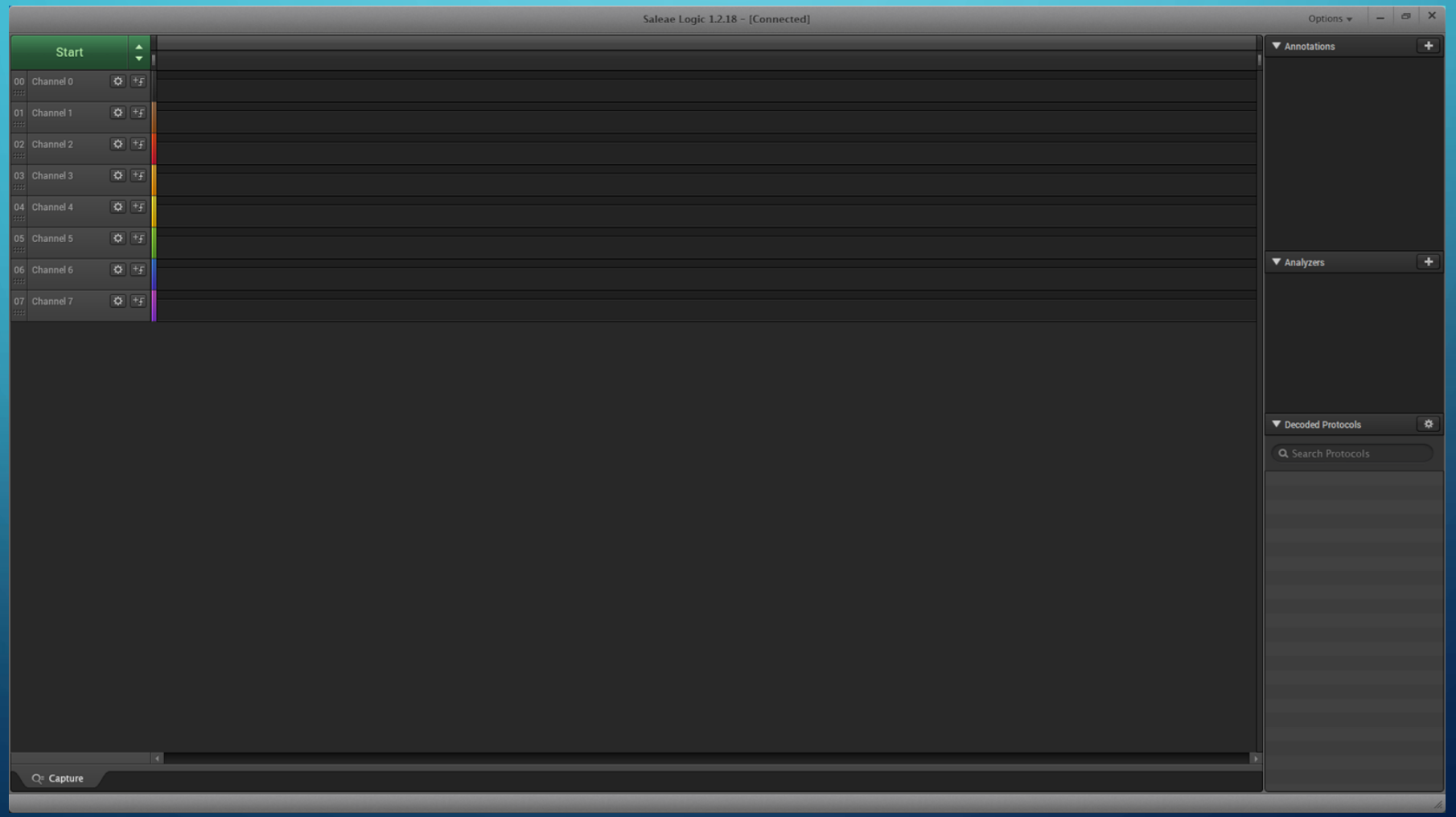




Pin1 Index Pointed By Arrow

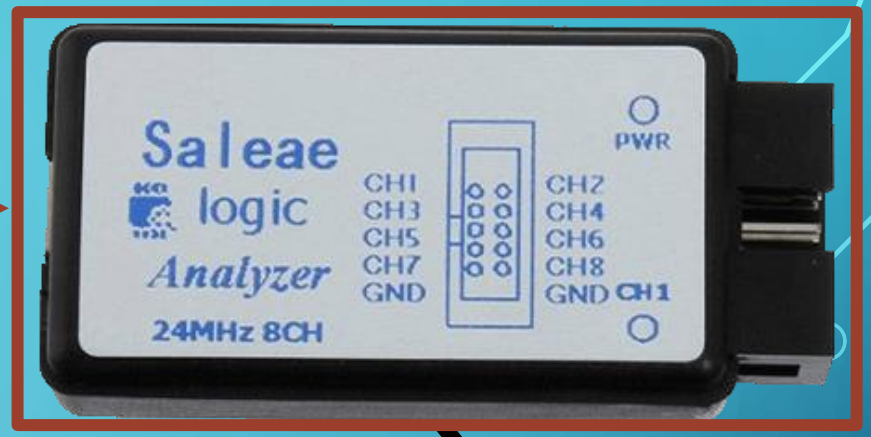
THE SOFTWARE INTERFACE

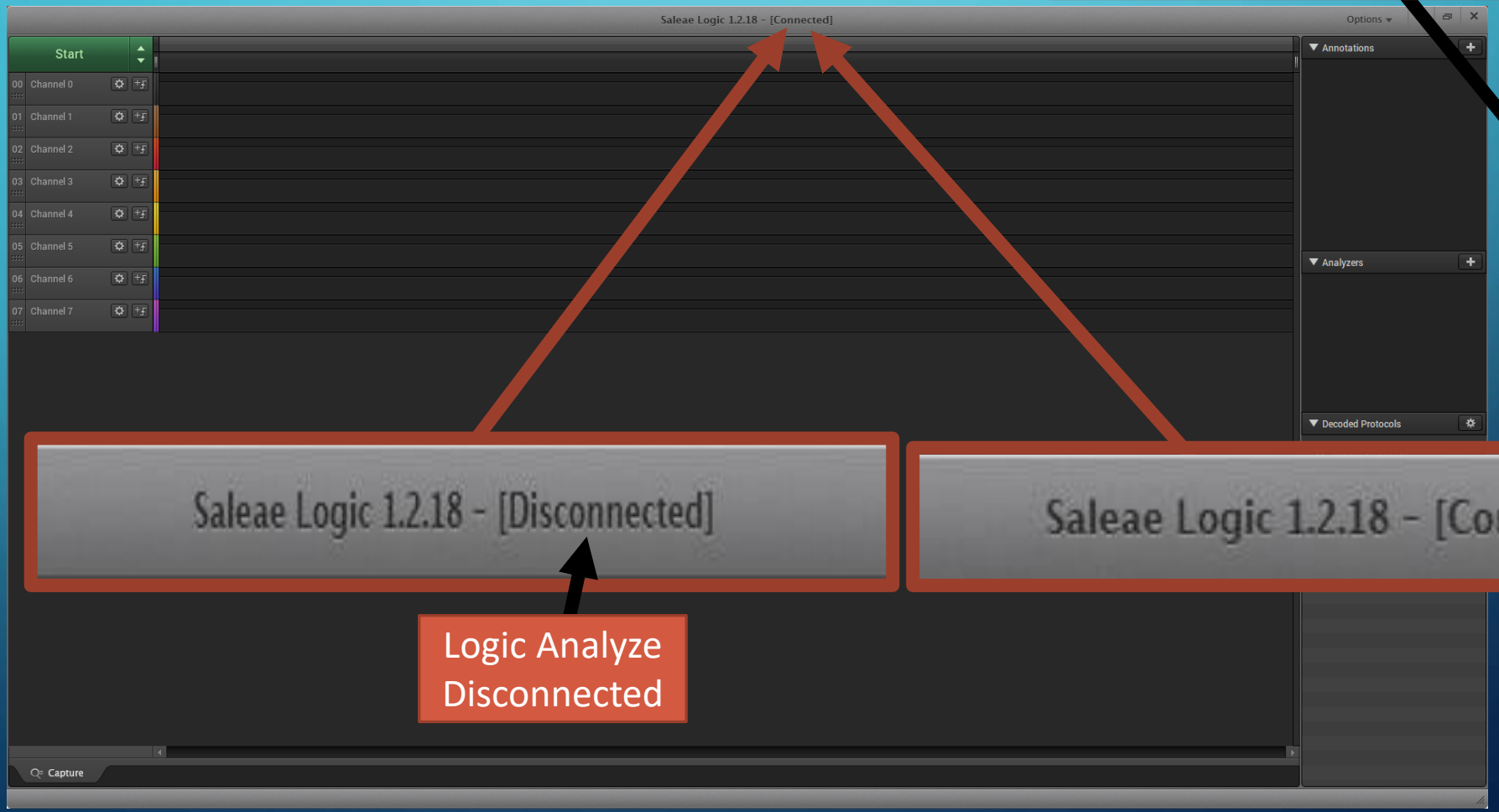
- Software Installing & How To Use:
  - Download the software Saleae Logic 1.2.18 and install it like any program.

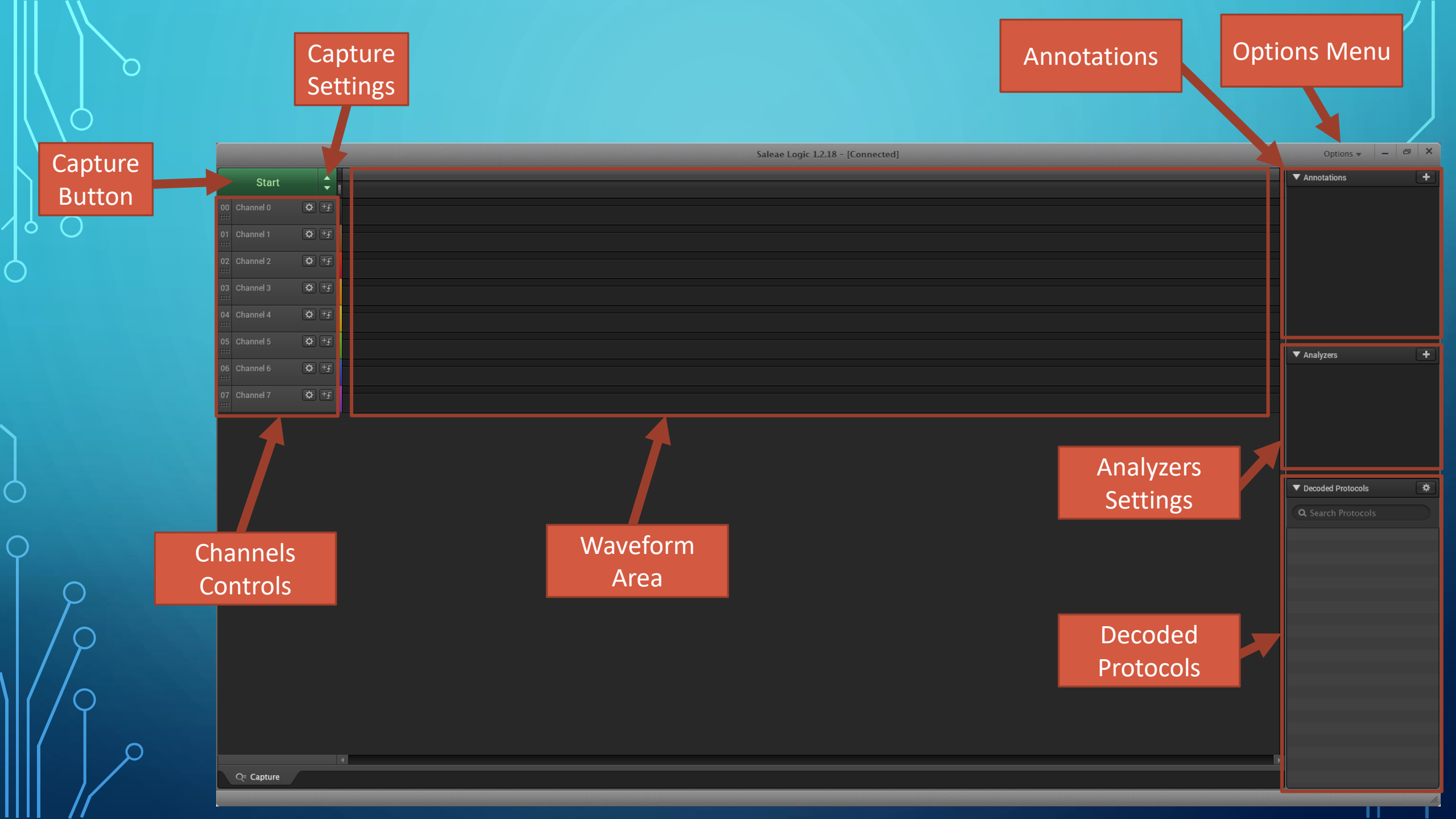


THE SOFTWARE INTERFACE

USB

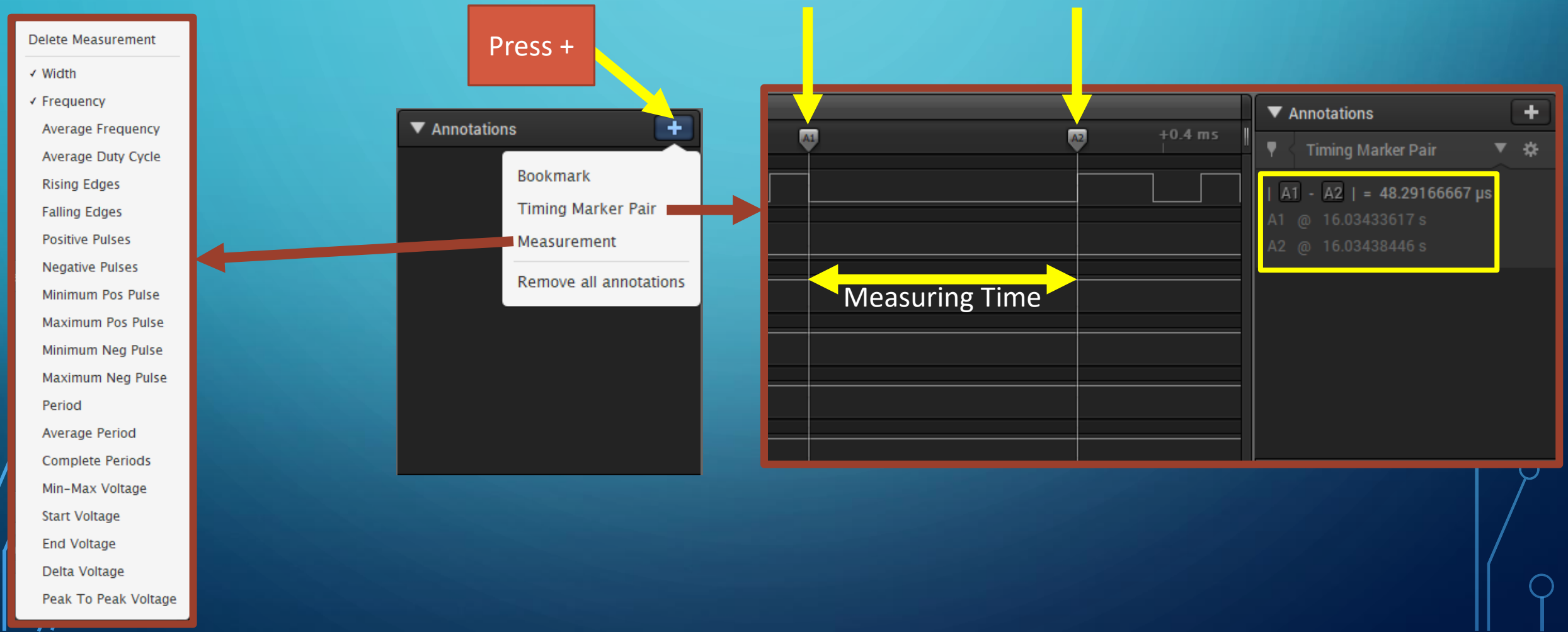





Capture Settings
Annotations
Options Menu
Capture Button
Saleae Logic 1.2.18 - [Connected]
Options
Start
00 Channel 0
01 Channel 1
02 Channel 2
03 Channel 3
04 Channel 4
05 Channel 5
06 Channel 6
07 Channel 7
Annotations
Analyzers
Decoded Protocols
Search Protocols
Capture
Channels Controls
Waveform Area
Analyzers Settings
Decoded Protocols

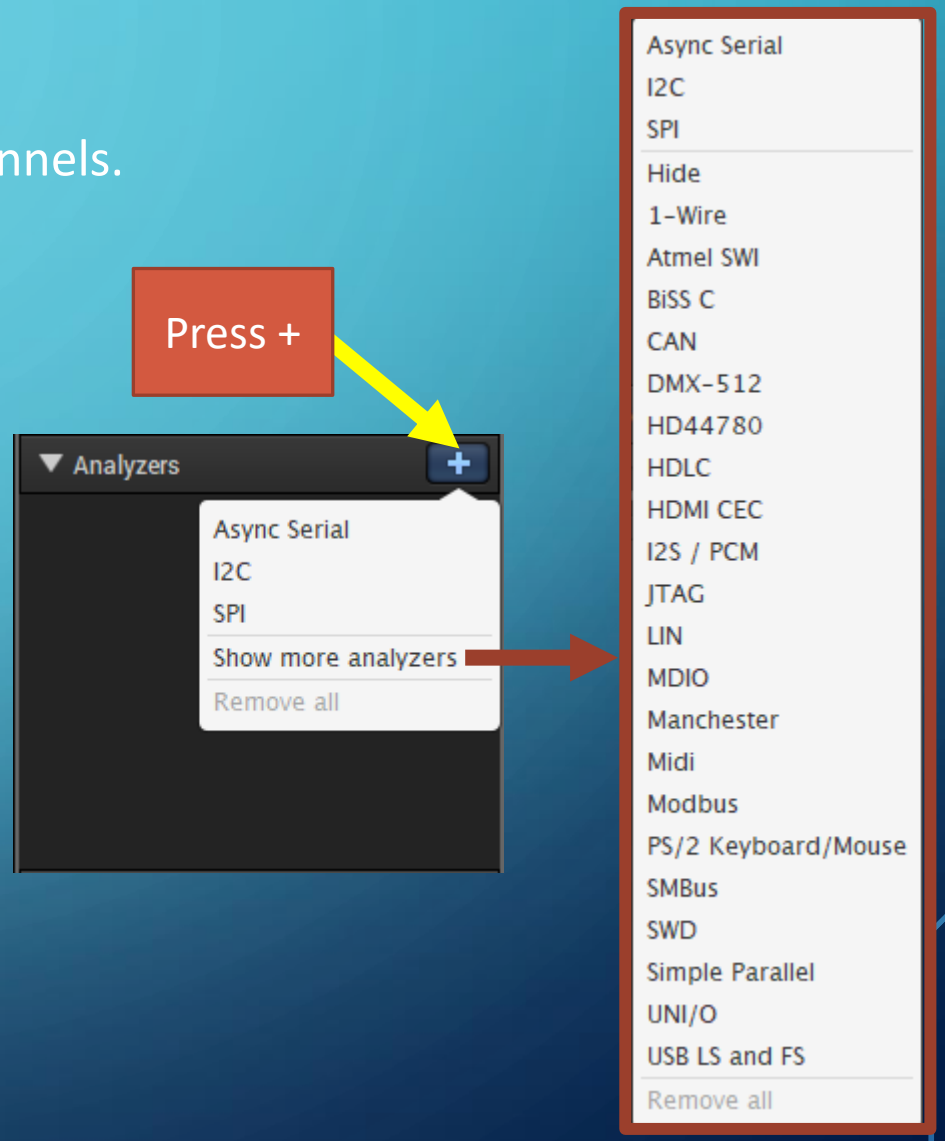
ANNOTATIONS

- Used for:
  - Adding Bookmarks, timing markers, and various measurements



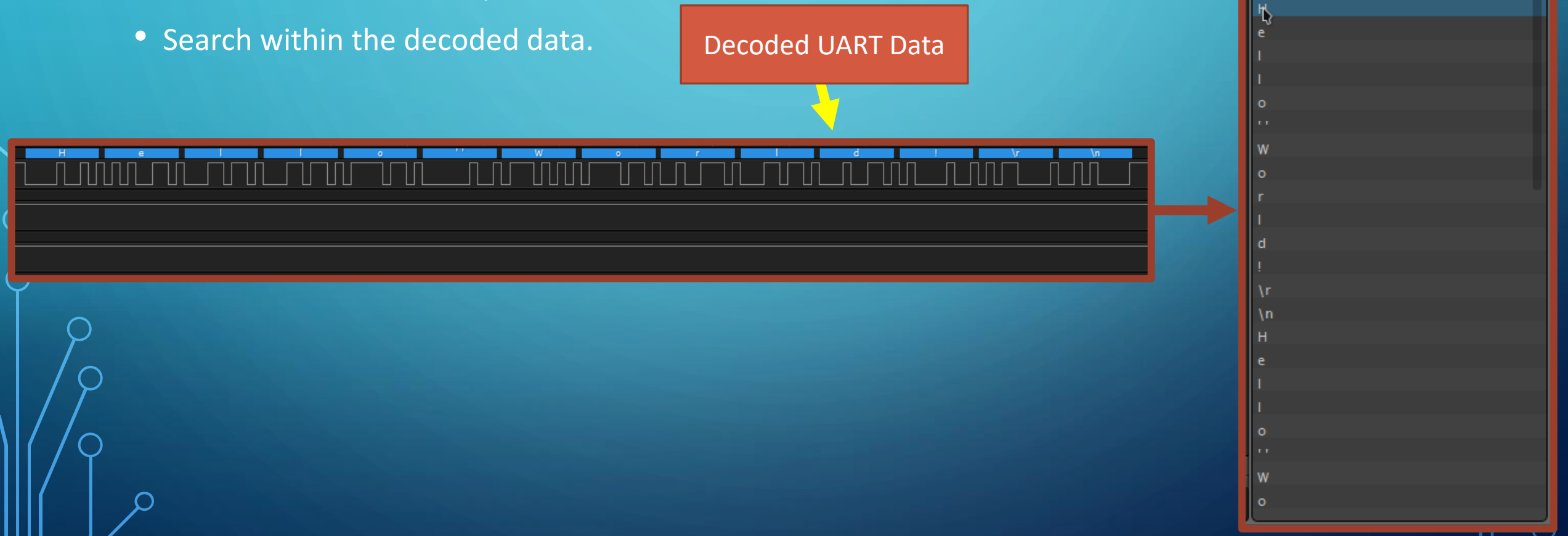
ANALYZERS

- Used for:
  - Adding one/many protocol analyzer(s) to the digital channels.



DECODED PROTOCOLS

- Used for:
  - Showing the decoded results of the protocol analyzers once the data has been processed.
  - Search within the decoded data.



CHANNELS CONTROLS

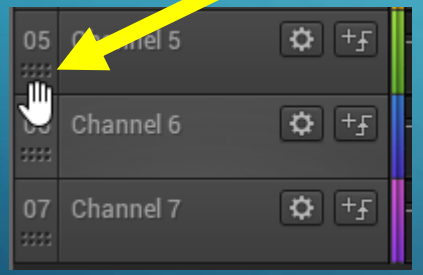
Decoded UART Data

- Used for:
  - Controlling the channels (Trigger, Zoom, Sort channels, Hide Channels)
  - Controlling the data capture (Sampling Rate, Sampling Duration)

CHANNELS CONTROLS

- Used for:
  - Controlling the channels (Trigger, Zoom, Sort channels, Hide Channels)
  - Controlling the data capture (Sampling Rate, Sampling Duration)

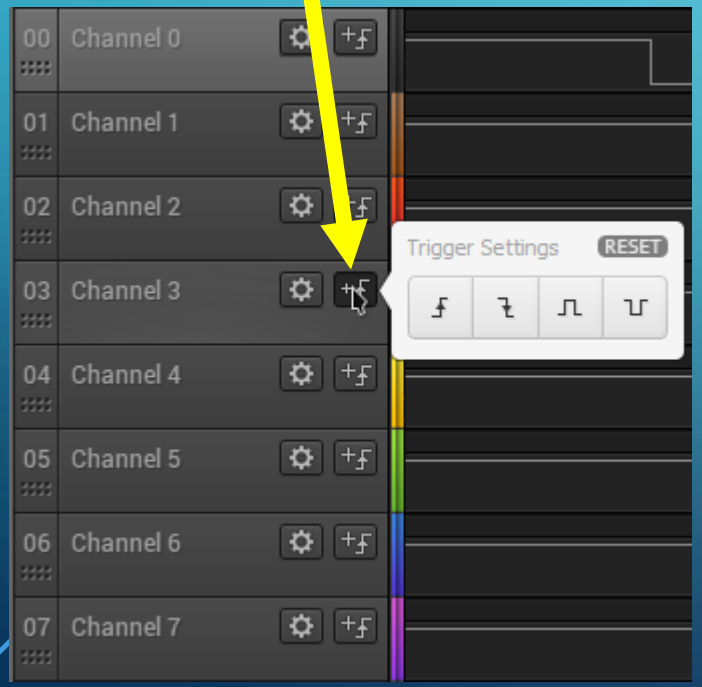
Hover on  and drag (up/down) to move the channel

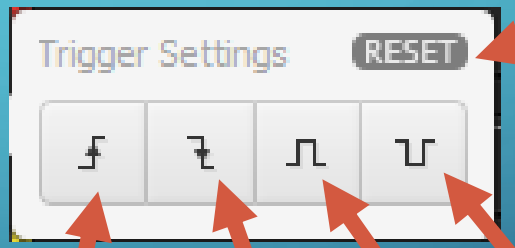


CHANNELS CONTROLS

- Used for:
  - Controlling the channels (Trigger, Zoom, Sort channels, Hide Channels)
  - Controlling the data capture (Sampling Rate, Sampling Duration)

Click on  to choose triggering settings for the channel





Reset Trigger (Remove Trigger)

Trigger on Rising Edge

Trigger on Falling Edge

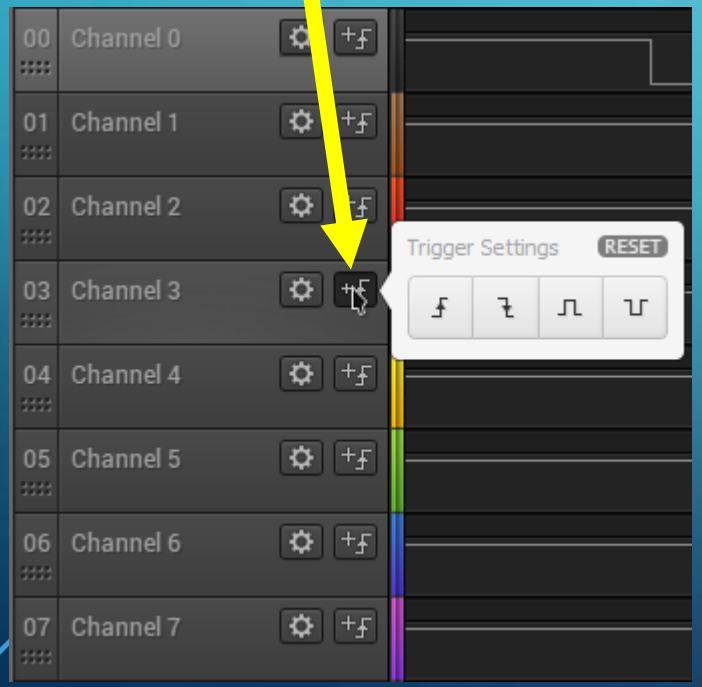
Trigger on Negative Pulse Width

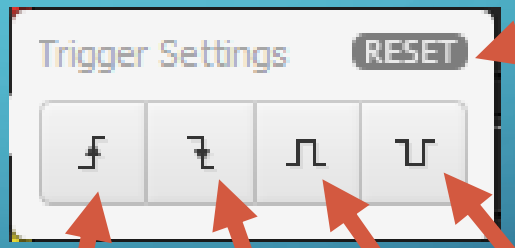
Trigger on Positive Pulse Width

CHANNELS CONTROLS

- Used for:
  - Controlling the channels (Trigger, Zoom, Sort channels, Hide Channels)
  - Controlling the data capture (Sampling Rate, Sampling Duration)

Click on  to choose triggering settings for the channel





Reset Trigger (Remove Trigger)

Trigger on Rising Edge

Trigger on Falling Edge

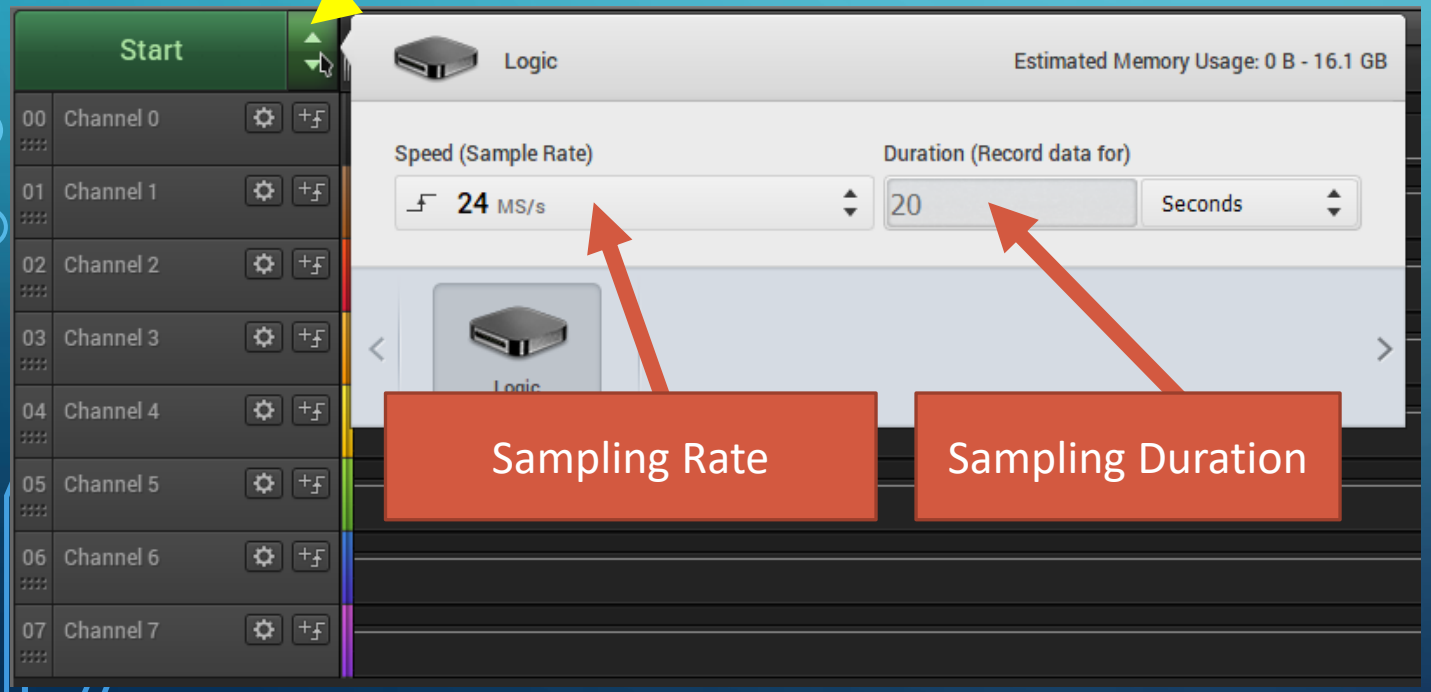
Trigger on Negative Pulse Width

Trigger on Positive Pulse Width

CHANNELS CONTROLS

- Used for:
  - Controlling the channels (Trigger, Zoom, Sort channels, Hide Channels)
  - Controlling the data capture (Sampling Rate, Sampling Duration)

Click on  to choose triggering settings for the channel

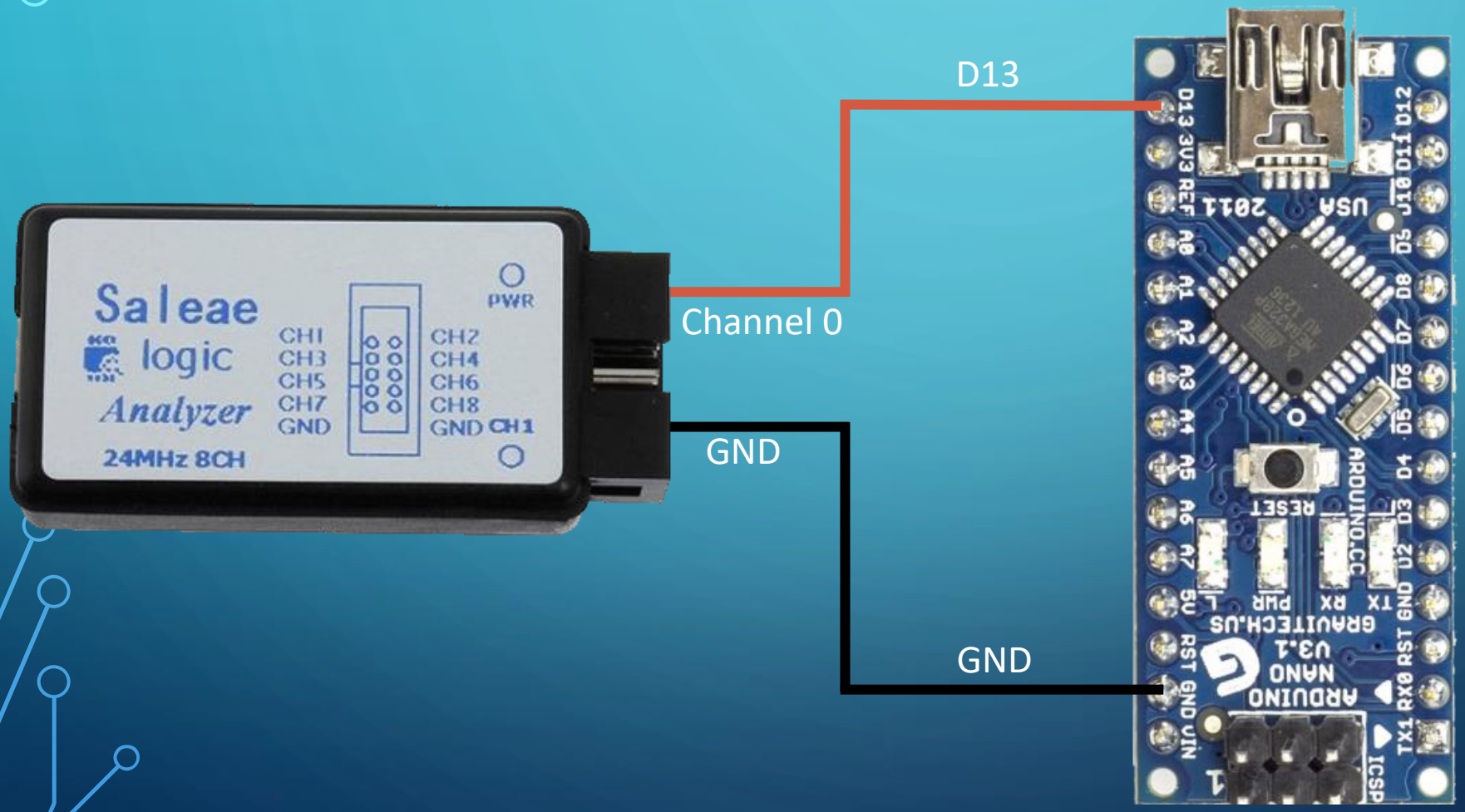


WHY USING ARDUINO IN THIS COURSE ?

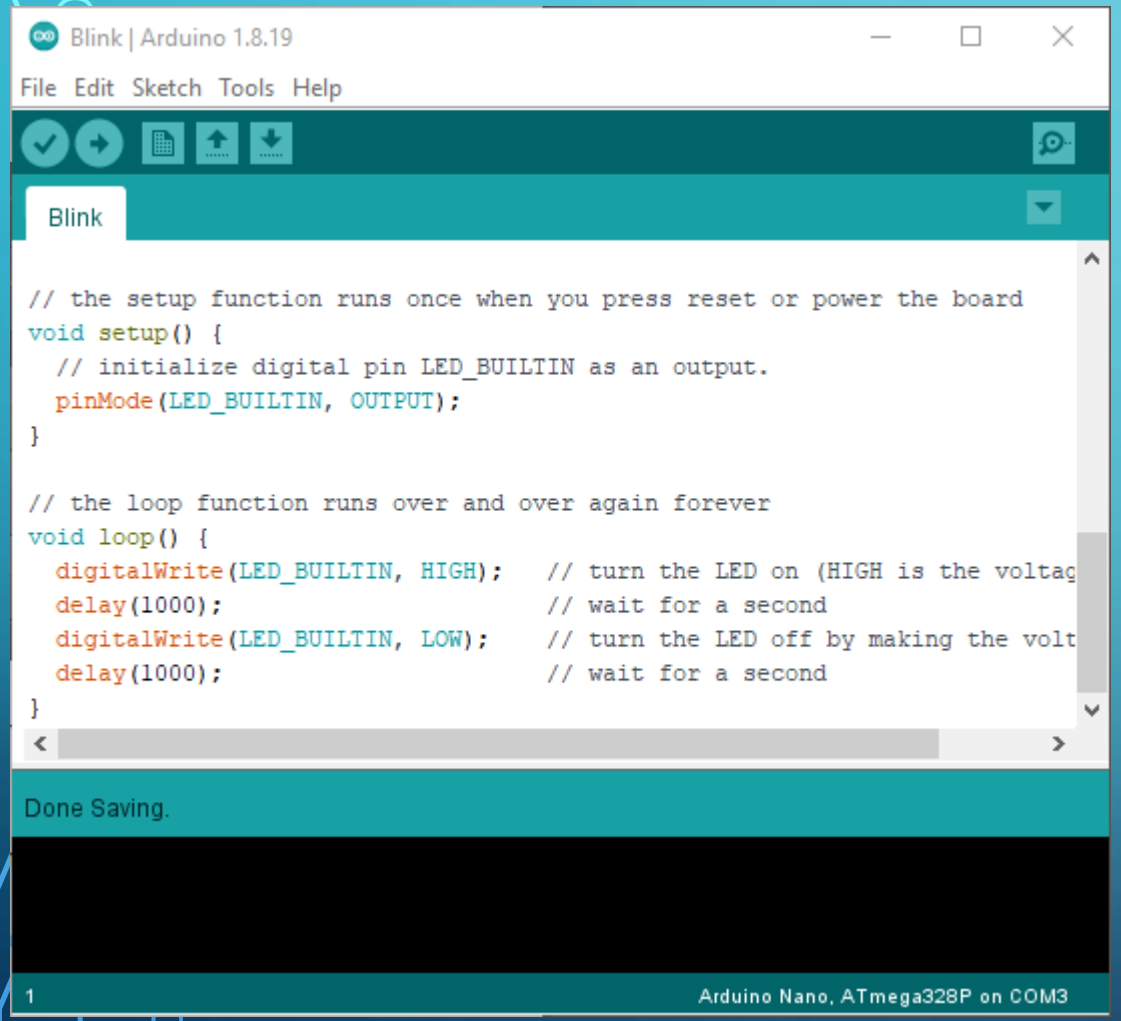
- In this course, I am interested in (Communication protocols Reverse Engineering). I will not write codes for communication protocols.
- Arduino is a simple prototyping platform with standard APIs.
- Arduino boards are everywhere, cheap, and affordable.

EXAMPLE 1 (LED BLINK)

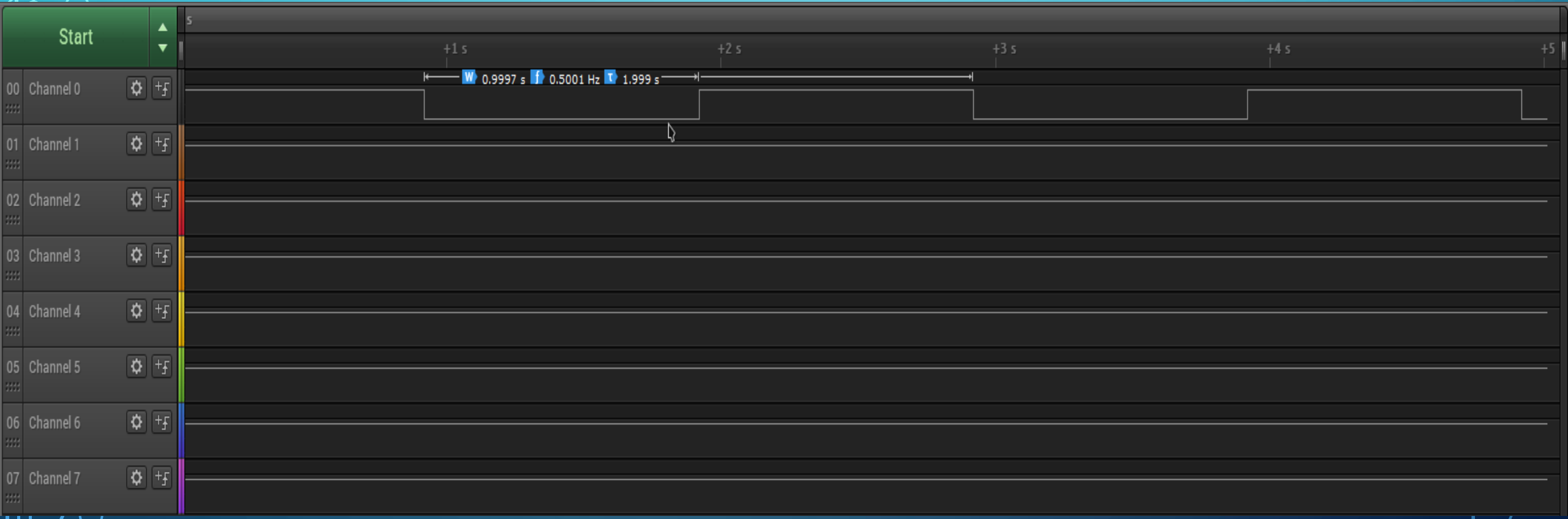
- This example blinks a LED every 1 second, lets record its GPIO data.



EXAMPLE 1 (LED BLINK)

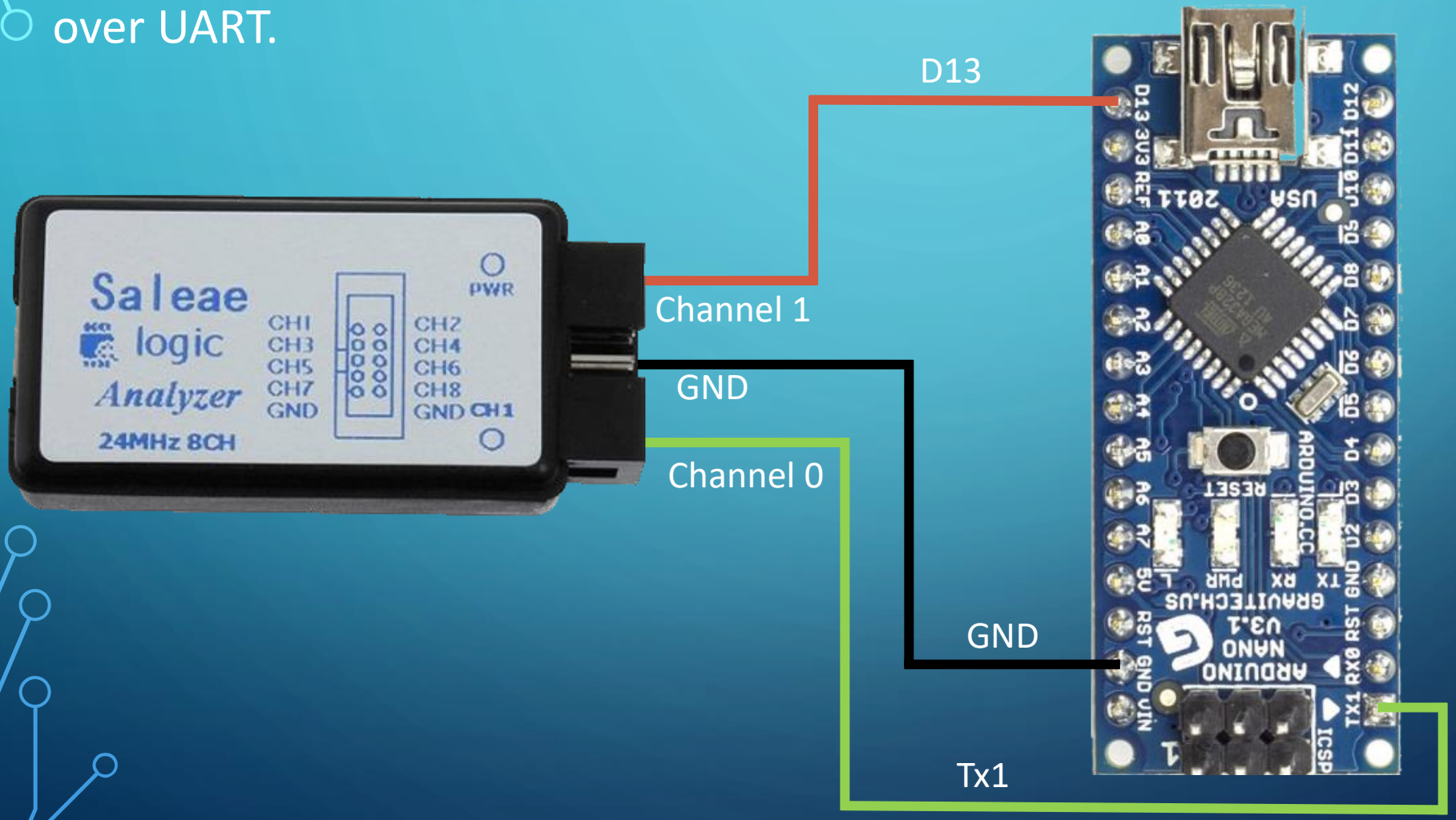


EXAMPLE 1 (LED BLINK)



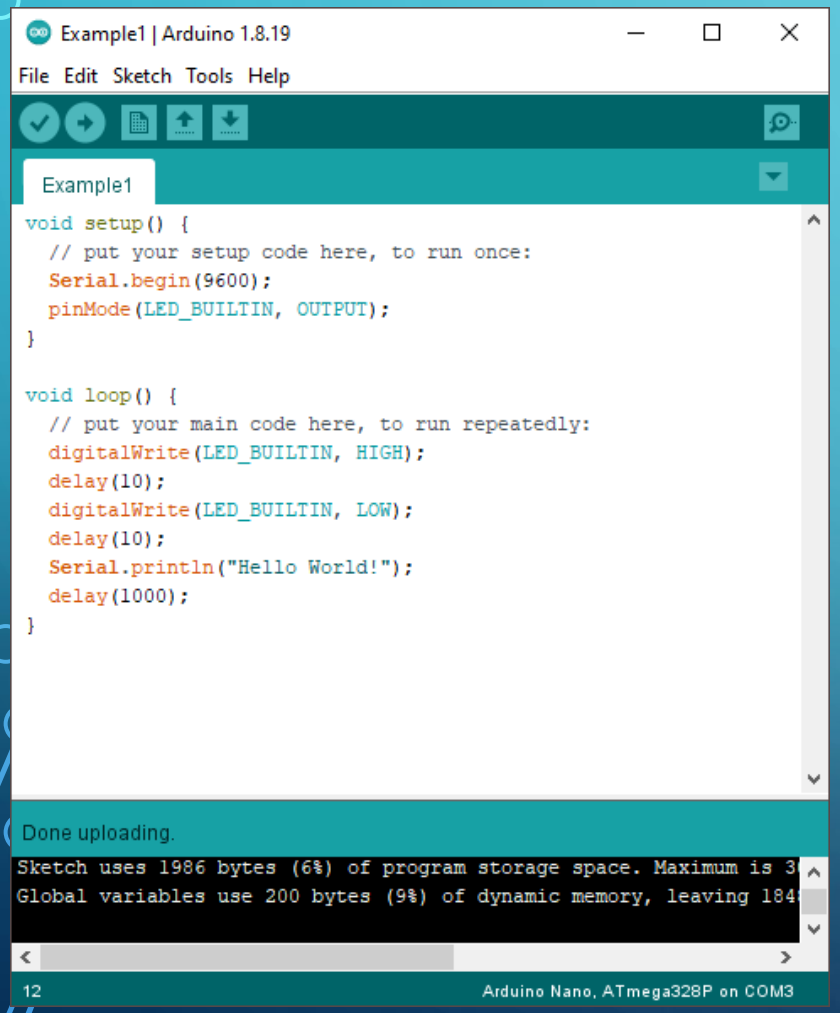
EXAMPLE 2 (LED BLINK + UART)

- This example generates a pulse with duration of 10ms on LED and sends “Hello World” over UART.



EXAMPLE 2 (LED BLINK + UART)

- This example blinks a LED every 1 second, lets record its GPIO data.



EXAMPLE 2 (LED BLINK + UART)

- This example blinks a LED every 1 second, lets record its GPIO data.