EECS 192: Mechatronics Design Lab

Discussion 3: Motor Driver and Servo Control

GSI: Andrew Barkan

February 3, 2020 (Week 3)

- Oscilloscope Basics
- Brief Code Segue

AD2 USB Oscilloscope Intro

- We can display a graph of voltage over time!
  - .... how is this useful?

Example scope capture

AD2 USB Oscilloscope Intro

- We can display a graph of voltage over time!
  - .... how is this useful?
- Provides visibility into your system
- Verify signals are what you expect:
  - Is your motor turning on?
  - Is your speed sensor outputting counts?
- Provide insights into subsystems:
  - See how line camera output works
- If you ever get stuck...
  - Don't debug by brute force
  - Turn on the scope and figure out the root of the problem

Example scope capture

WaveForms Software

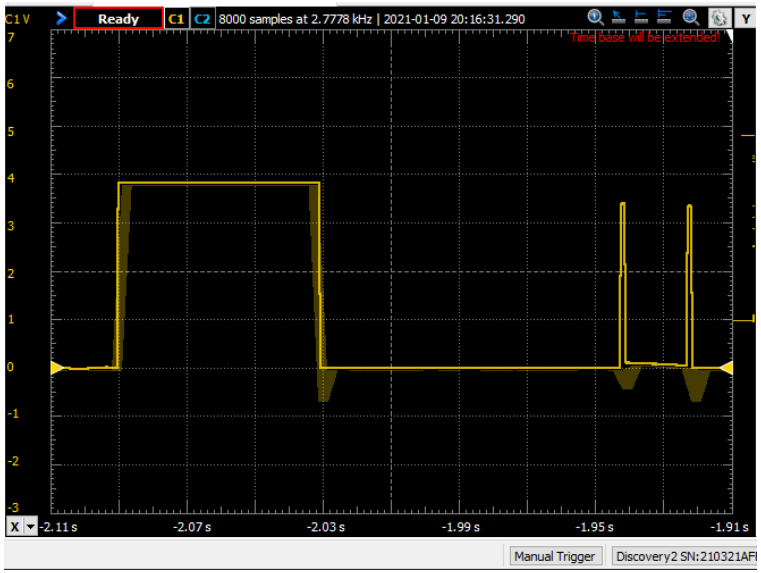
Do you have WaveForms installed?

- Make sure your AD2 is connected
- Then start WaveForms
- WF will let you know if device is not recognized
- If you don't have an AD2, you can still follow along in demo mode!

blog.digilentinc.com

Viewing Window

- You should always adjust to fit your data
  - very possible you'll scope something you don't want to
  - or have no idea what you're looking at
- Know how to manually set the scope
  - You should know what to expect
  - Set the per-channel vertical scale based on the expected voltage range
  - Set the global horizontal scale based on expected timescale



ESC startup in factory condition

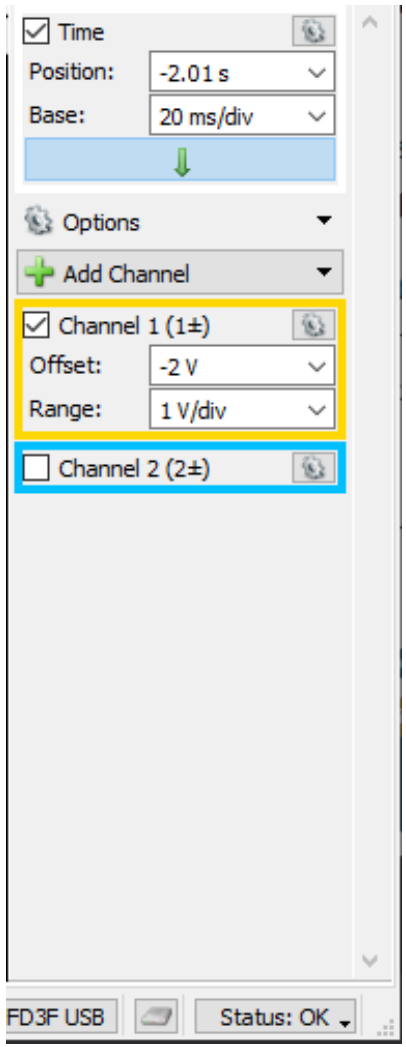
You have two different channels (1 and 2)
Each channel will display as separate color in window unless deselected

- Vertical = Voltage
- Horizontal = Time

Adjustments

- Scaling per division (i.e. s/div, V/div)
- Voltage offset
- Position in time

Lots of other functions that we won't go into here, but feel free to explore on your own!



Scaling and channel options

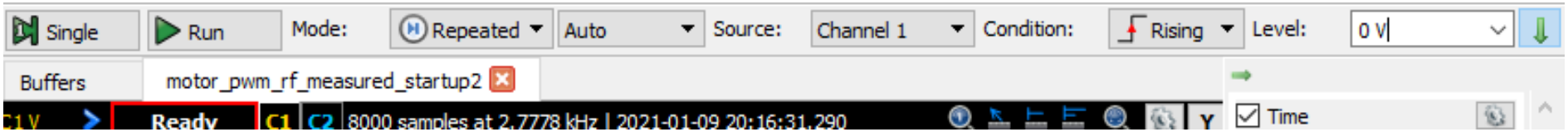
Triggering

- Triggering functions allow us to capture signal edges for inspection
- Useful for tracking signals like PWMs

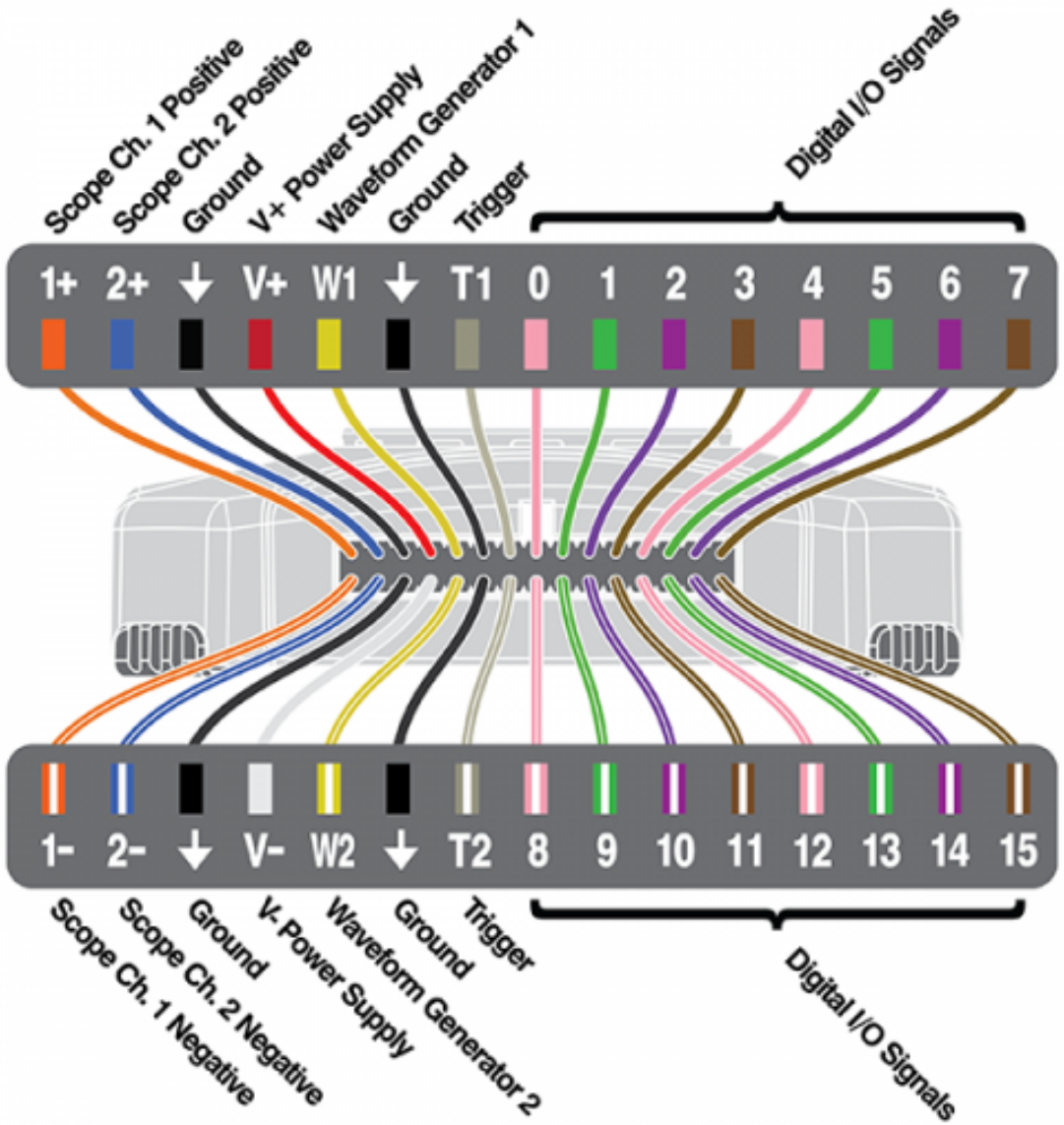
Adjustments

- Channel for Triggering
- Type of scanning
- Trigger condition (rising, falling, either)
- Level for trigger (V)

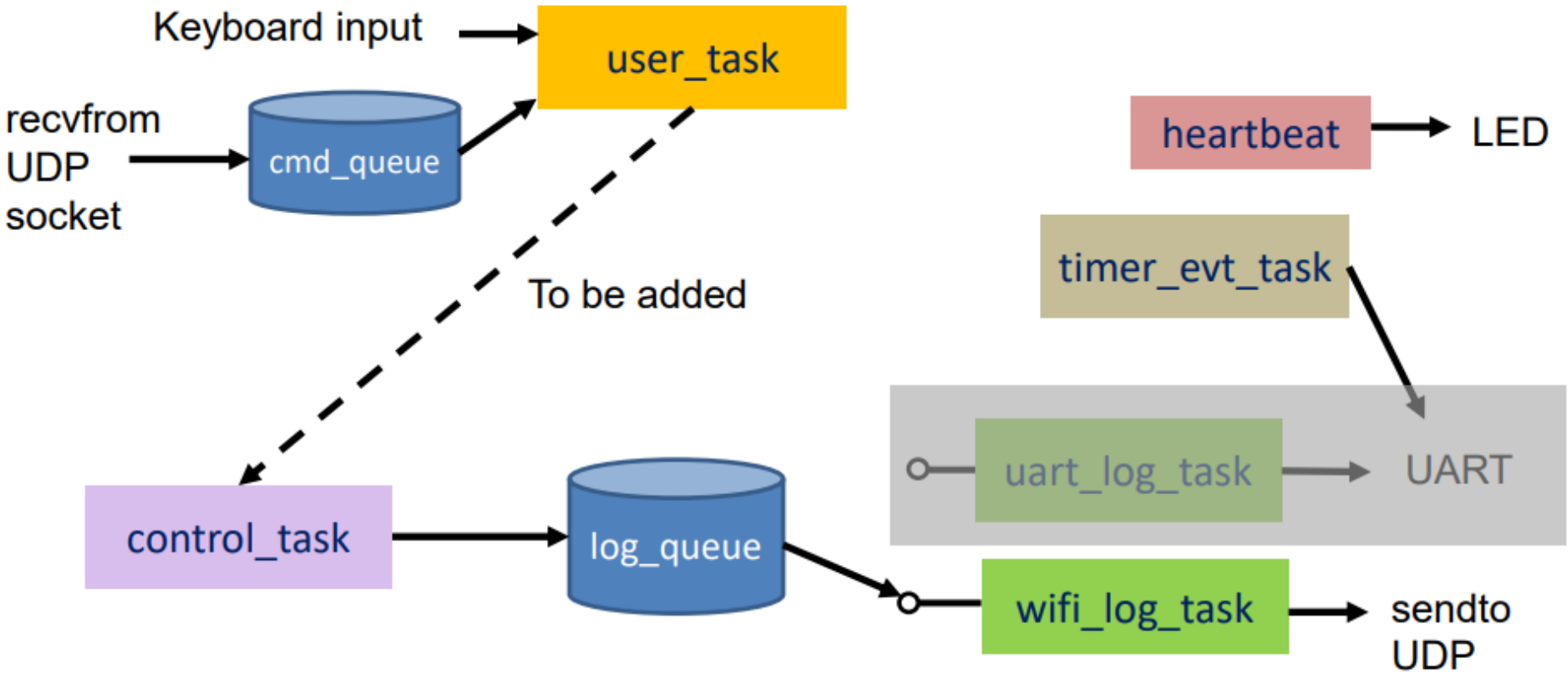
Either run continuously or trigger single acquisition sets



AD2 Pinout


Scope Ch. 1 Positive
Scope Ch. 2 Positive
Ground
V+ Power Supply
Waveform Generator 1
Ground
Trigger
Digital I/O Signals
1+ 2+ V+ W1 T1 0 1 2 3 4 5 6 7
1- 2- V- W2 T2 8 9 10 11 12 13 14 15
Scope Ch. 1 Negative
Scope Ch. 2 Negative
Ground
V- Power Supply
Waveform Generator 2
Ground
Trigger
Digital I/O Signals

New UDP Command Interface



Designing Commands

Let's say we want to take our code from last checkpoint (PWM fade) and control the frequency

- This function lives in our control task
- Need to define our own command set that makes use of existing infrastructure!
- Use the same format as the skeleton code
- Question: what is the **safest** way to exchange info between tasks in FreeRTOS?

Designing Commands

Let's say we want to take our code from last checkpoint (PWM fade) and control the frequency

- This function lives in our control task
- Need to define our own command set that makes use of existing infrastructure!
- Use the same format as the skeleton code
- Question: what is the **safest** way to exchange info between tasks in FreeRTOS?
- A queue!

Let's see an example of a control command in the code!

Then we will inspect with our oscilloscope!