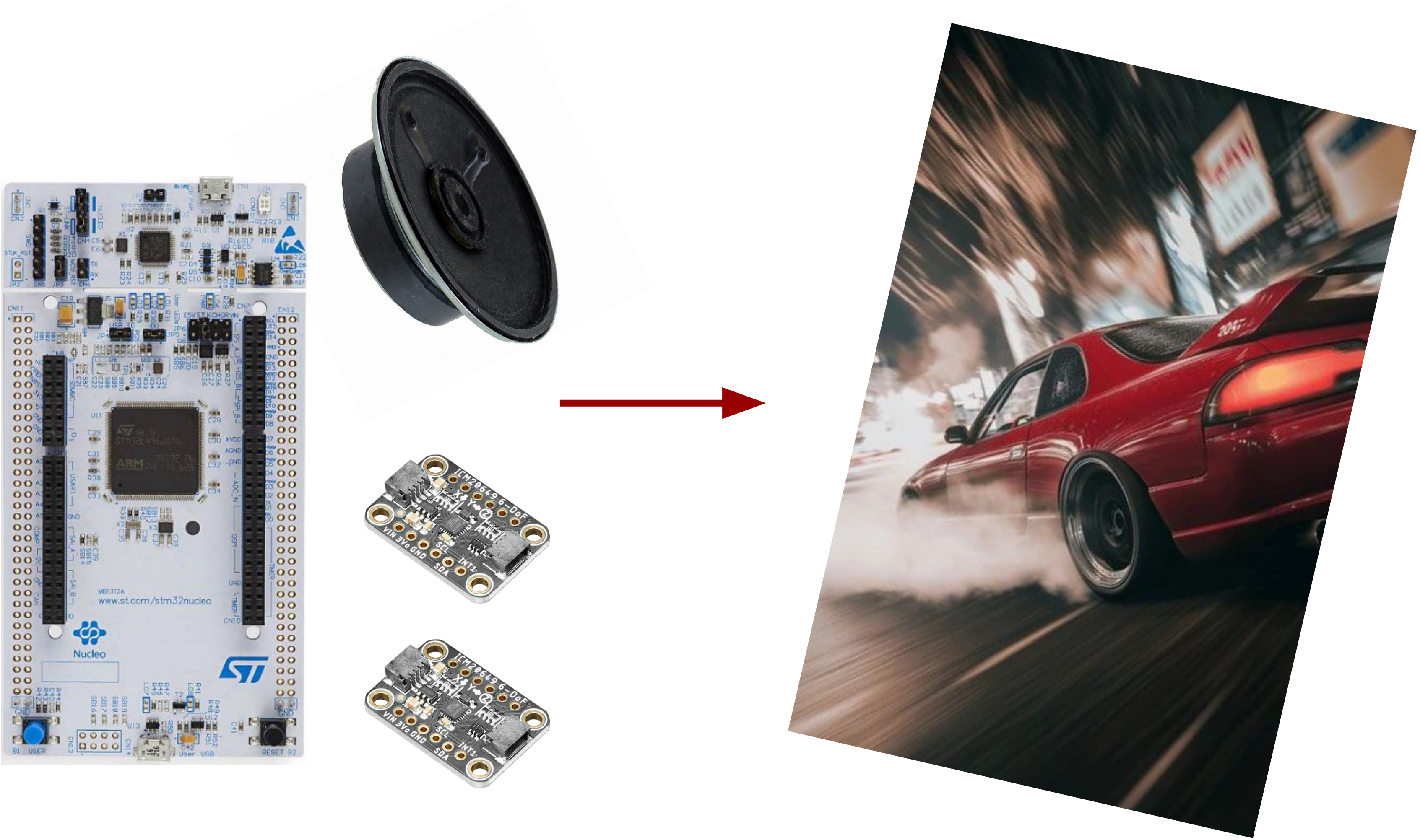
Angel Santana, Ben Otter,
Chris Lann, Luis Sanchez


www.st.com/stm32nucleo
Nucleo
ICM20649 6-DoF
VIN 3Vo GND SCL SDA INT1
ICM20649 6-DoF
VIN 3Vo GND SCL SDA INT1

Bicycle Drift

Angel Santana, Ben Otter, Chris Lann, Luis Sanchez

Motivation

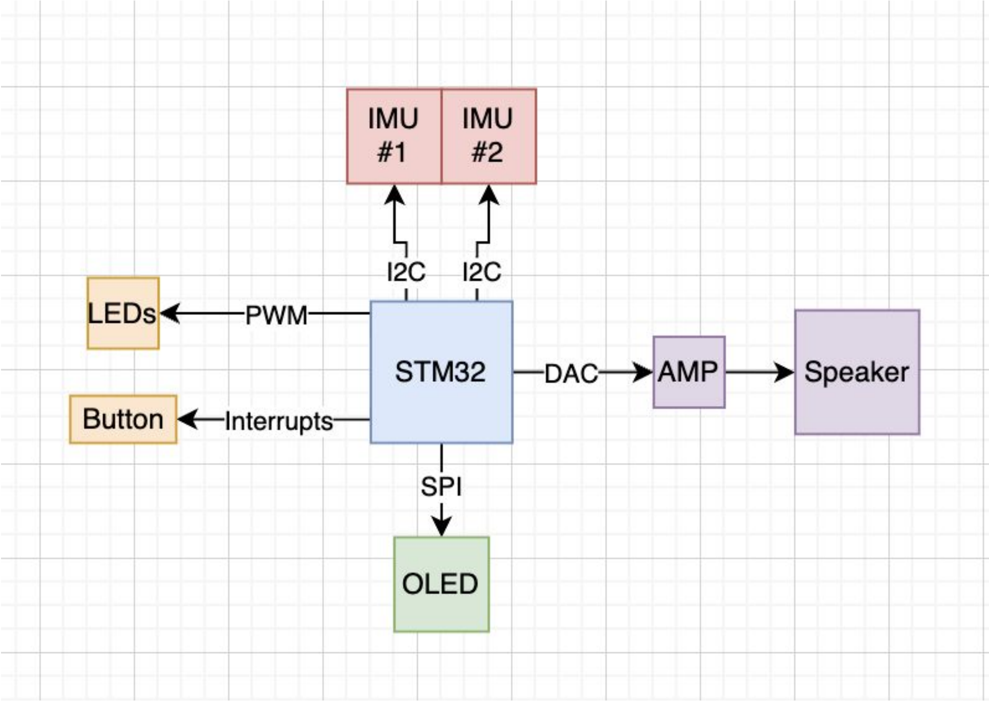
- Riding a regular bike is boring
  - Wouldn’t it be cool with sound effects, especially dramatic drifting noises.
- Existing bike accessories are mostly lights, bells, or a basket where the exciting sound effects?

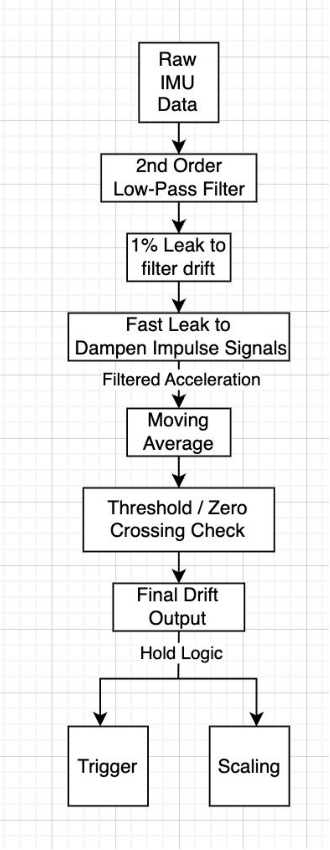
Solution

- Develop a compact, drift detection module to detect drift events, and provide visual, audio, and on-screen feedback
  - Uses dual IMUs to measure centripetal acceleration
  - Detects drift events in real time
  - Responds with sound, LEDs
  - Button control for user interface with on-screen feedback

- Imagine how cool it would feel to have this on your bike as you go through the roundabouts by main quad at 1:30pm!!!!!!

Block Diagram of Embedded System





Implementation overview

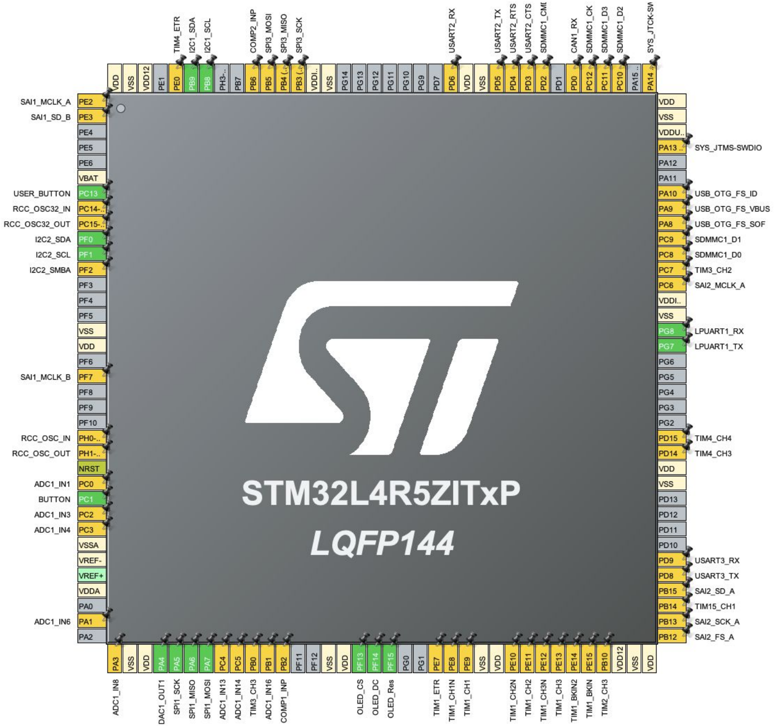
- Button:
  - Software debouncing
  - Single press, double press, long press detection
  - Triggers Interrupt to cycle through OLED screens
- LEDs:
  - Drift magnitude controls brightness through PWM
- OLED:
  - 4 wire SPI, full-duplex
  - Display acceleration data, sound selection, volume bar
- Audio:
  - PAM 3802 Audio Amplifier & Speaker
  - Convert .MP3 -> .WAV and output data to DAC
- Accelerometers:
  - 2x LSM303AGR Accelerometers as gyroscope

Implementation code overview

*Button external interrupt, DMA transfer for audio buffer

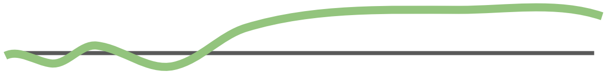
Main while loop:

1. Handle button detection
2. Poll accelerometers and update filter
3. Refresh display if necessary
4. Start DMA transfer for audio playback when turn detected

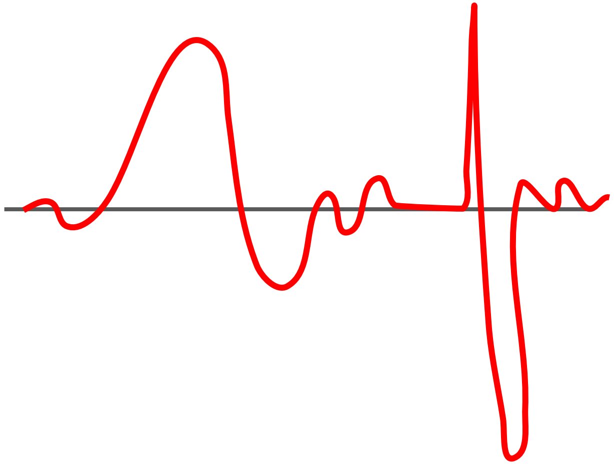


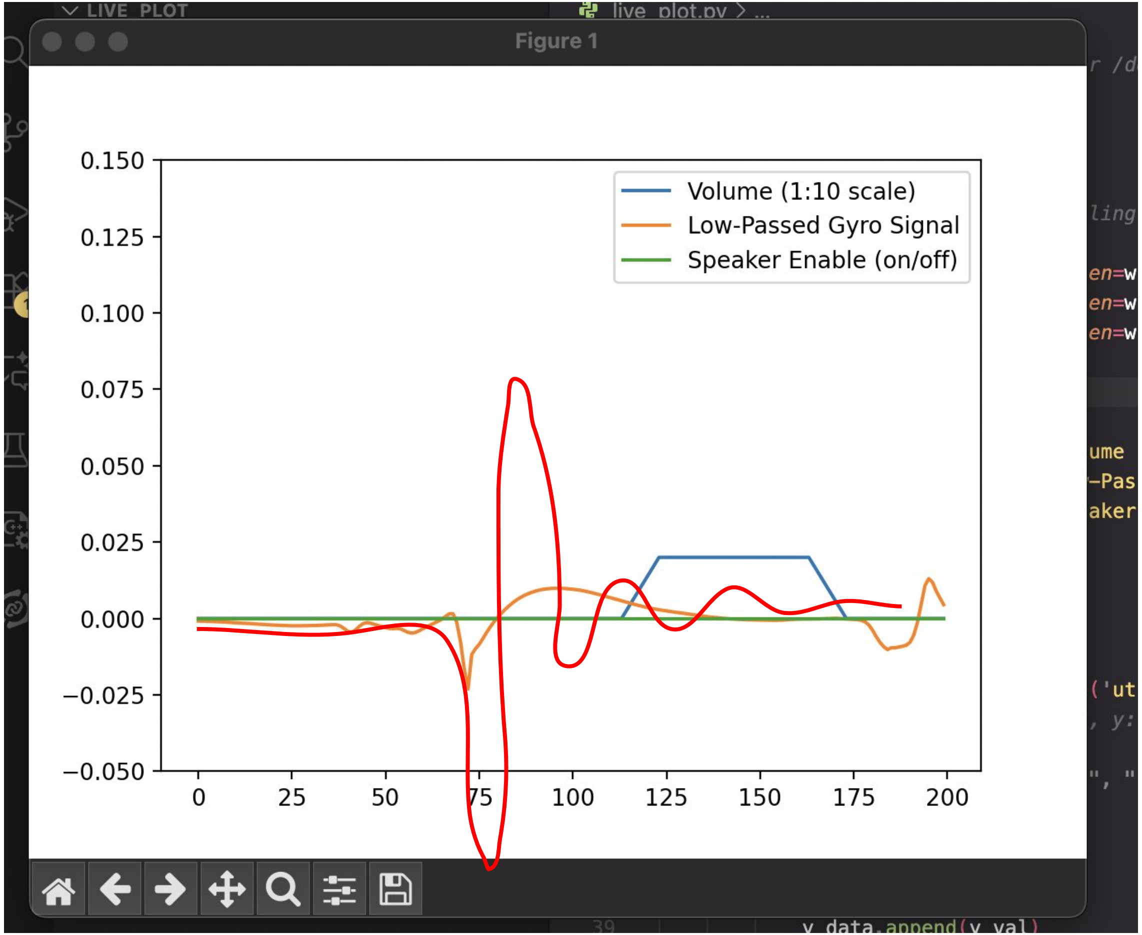
Data Filtering

✓

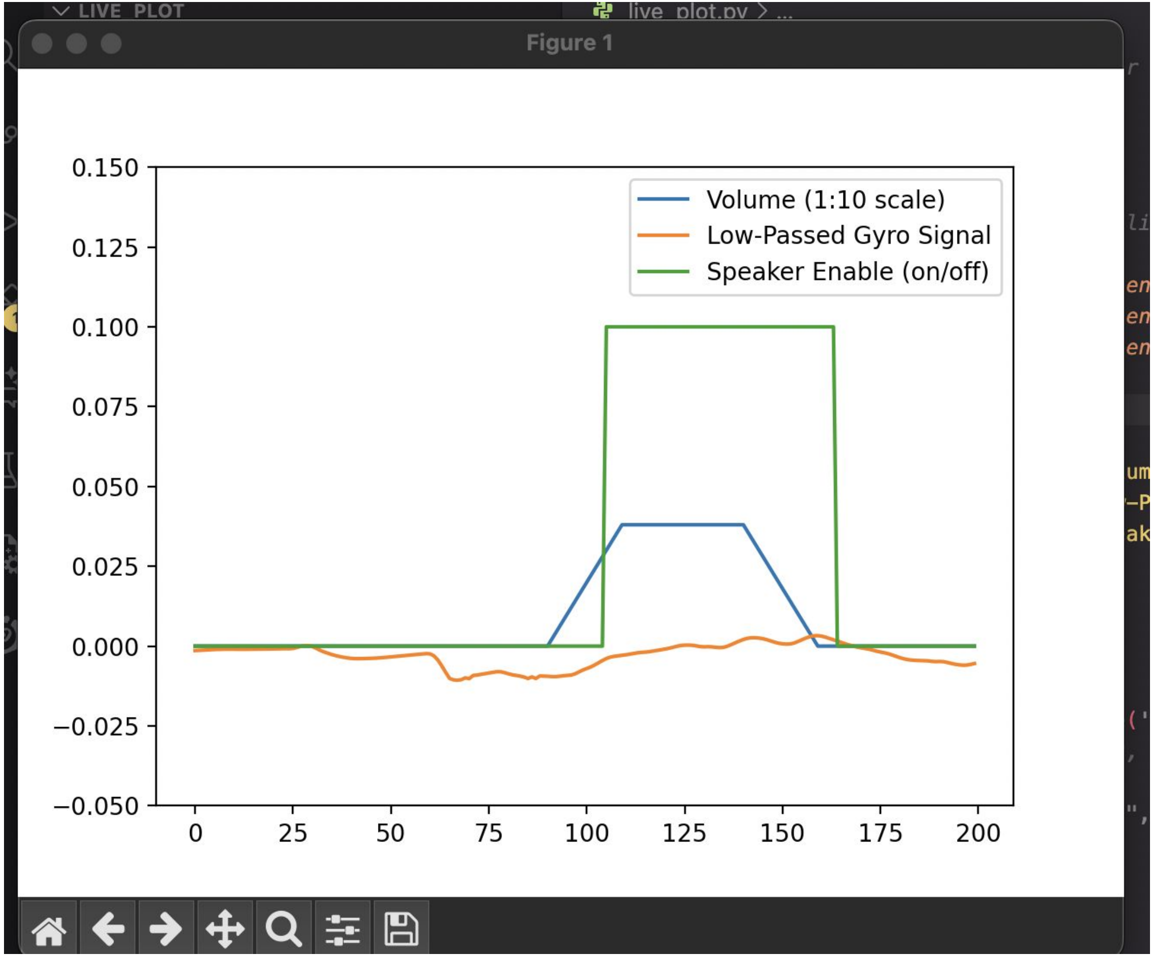


✗





```
y_data.append(y_val)
```



LIVE_PLOT
live_plot.py > ...
Figure 1
0.150
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
0
25
50
75
100
125
150
175
200
Volume (1:10 scale)
Low-Passed Gyro Signal
Speaker Enable (on/off)

Implementation challenges

1. Loading audio samples into memory
2. Volume scaling
3. Button detection
4. Full system integration
5. Real time filtering
6. Powering everything from 9V battery
7. Modularizing the code for easy system integration

Results