

farbot's RealtimeMutatable

```
using FrequencySpectrum = std::array<float, 512>;  
RealtimeMutatable<FrequencySpectrum> mostRecentSpectrum;  
  
void processAudio (const float* buffer, size_t n) {  
    auto& freqSpec = mostRecentSpectrum.realtimeAcquire();  
    freqSpec = calculateSpectrum (buffer, n);  
    mostRecentSpectrum.realtimeRelease();  
}  
  
void updateSpectrumUIButtonClicked() {  
    auto& recentSpectrum = mostRecentSpectrum.nonRealtimeAcquire();  
    displaySpectrum(recentSpectrum);  
    mostRecentSpectrum.nonRealtimeRelease();  
}
```

farbot's RealtimeMutatable

```
using FrequencySpectrum = std::array<float, 512>;

RealtimeMutatable<FrequencySpectrum> mostRecentSpectrum;

void processAudio (const float* buffer, size_t n) {
    RealtimeMutatable<FrequencySpectrum>::ScopedAccess<true> freqSpec(mostRecentSpectrum);
    *freqSpec = calculateSpectrum (buffer, n);
}

void updateSpectrumUIButtonClicked() {
    RealtimeMutatable<FrequencySpectrum>::ScopedAccess<false> recentSpectrum(mostRecentSpectrum);
    displaySpectrum(*recentSpectrum);
}
```