

オープンソースカンファレンス2021 Nagoya
Jetson Nano超入門 + α
2021年5月29日

Jetson NanoでSDR (Software Defined Radio)

古瀬 勉

自己紹介

名前：古瀬 勉（ふるせ つとむ）

職業：

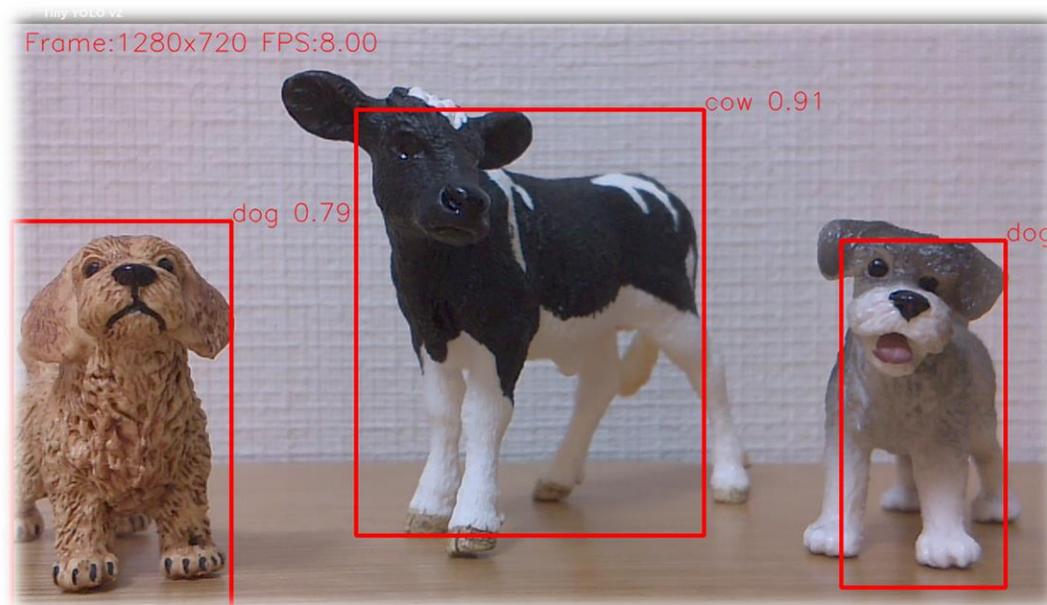
半導体商社のフィールド・アプリケーション・
エンジニア（FAE）



趣味と仕事の両面で、Jetson Nanoに取り組んでいます

執筆パート

Part 5 USBカメラを使った物体検出 (Chapter 5-5を除く)

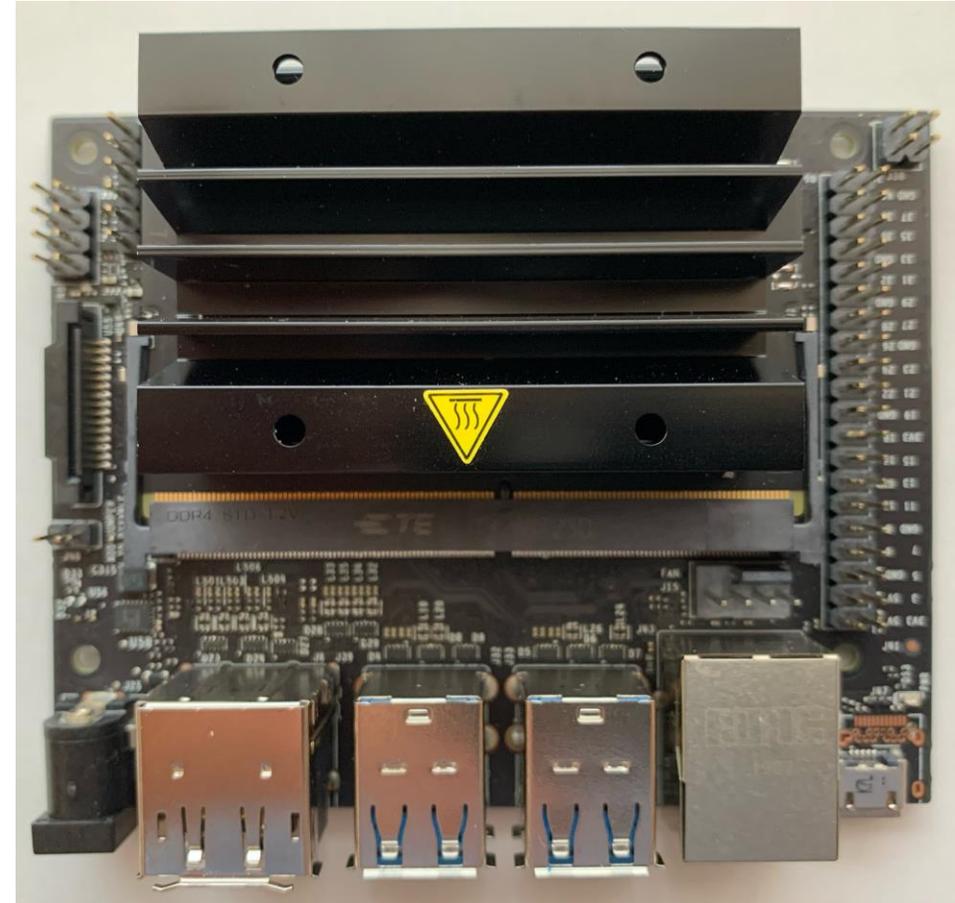


Part 9 Dockerで簡単デプロイメント 改訂第2版で新規追加



Jetson Nanoでは 128コアのGPUが使える！

- **GPU**
 - **128-core NVIDIA Maxwell™ GPU**
- **CPU**
 - Quad-Core ARM® Cortex®-A57 MPCore processor
- **AI Performance**
 - 472 GFLOPs



GPUによる並列処理は デジタル信号処理にも最適

- GPUはデータ並列化に関する処理が得意
 - 同時に処理できるデータが大量にあるとき、それらのデータを各コアに分散させる
- デジタル信号処理もデータ並列化で処理できる場合が大半
 - 高速フーリエ変換（FFT）、畳み込み、フィルタなど
- [RAPIDS cuSignal](#)と[CuPy](#)でNVIDIA CUDAプログラミング・モデルに基づくGPUの利用が可能（プログラミング言語はPython）
 - **cuSignal** **Scipy Signal**互換API
 - **CuPy** **NumPy**互換API

Software Defined Radio : SDR

- ハードウェアを変更せずに制御ソフトウェアにより、無線通信方式を変更できる技術
- 安価なUSB Dongle型の受信機で実験できる
 - 受信機は高周波処理まで、変調は行わない
 - ARM Linux対応のドライバも提供されているので、Jetsonでも動作



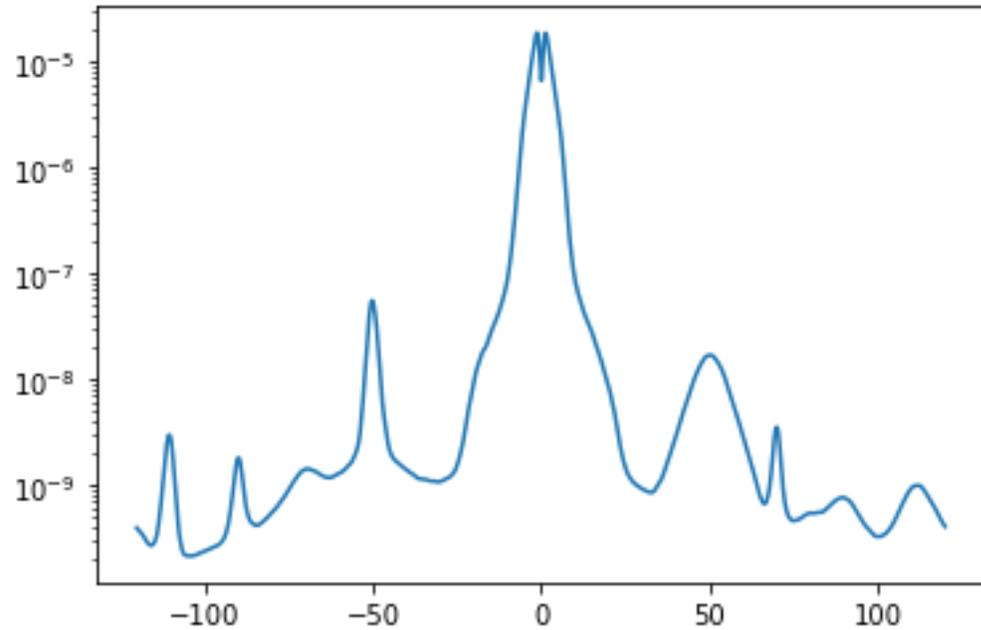
FM放送のデジタル復調を Pythonで書ける

- IQ表現→位相角→位相角のシフト→微分→ダウンサンプリング
→音声

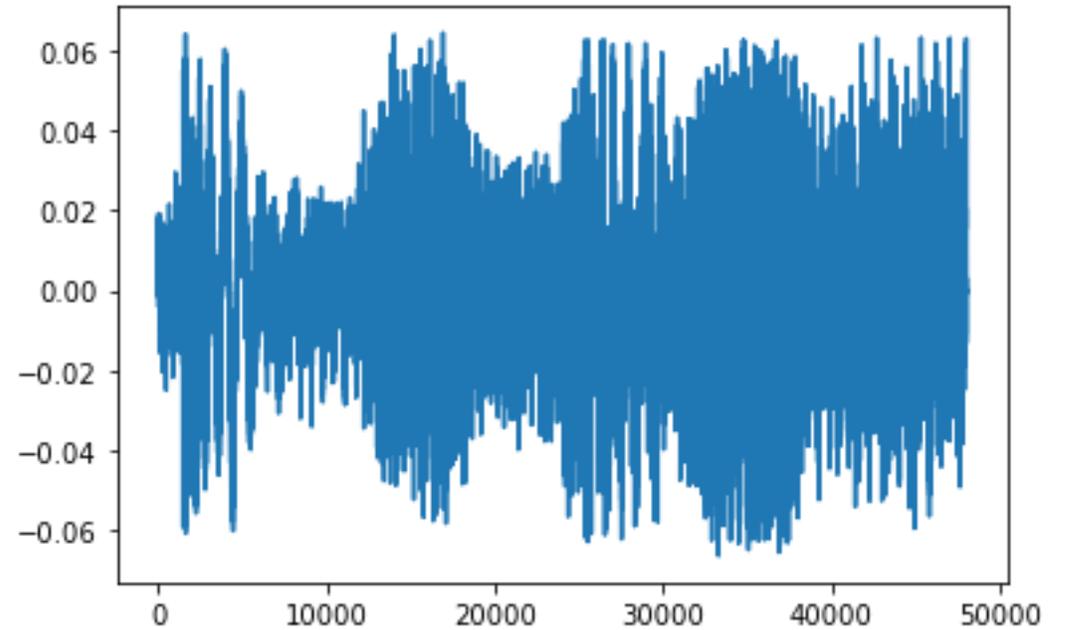
```
def demodulate(buff, resample_factor):  
    b = buff  
    b = np.diff(np.unwrap(np.angle(b)))  
    b = sig.resample_poly(b, 1, resample_factor,  
        window='flattop')  
    b /= np.pi  
    return b
```

処理結果の可視化

パワースペクトルの表示



復調された波形



GPUによる高速化（FM復調）

音声1秒間分のFM復調処理（10回計測した平均値）

	CPU Time User [msec]	CPU Time Sys [msec]	CPU Time Total [msec]	Wall Time [msec]
CuPy + cuSignal GPU利用	13.6	2.4	16.0	126.0
NumPy + SciPy Signal CPUのみ	530.0	86.8	616.0	619.0

SDRとAIの組み合わせ

- 今後は、SDRとAIの組み合わせが期待される。
たとえば、
 - スペクトラムをAIでモニターして、最適なチャンネルへ、送信側を誘導
 - 最適な変調方式を、送信側へ指示
 - 音声認識

詳細はQiitaでご覧ください

- <https://qiita.com/tsutof>
- [NVIDIA Jetson Nano 開発者キットでソフトウェア無線
\(セットアップ編\)](#)
- [NVIDIA Jetson Nano 開発者キットでソフトウェア無線
\(動作確認編\)](#)