

# ISGK INSTRUMENTS アイ・エス・ジー・ケー・インストゥルメンツ

## Digital Synth VRA8-N for Arduino Uno

### 2018.08.04-05 MAKER FAIRE TOKYO 2018

Arduino Uno の限界に挑戦する「Digital Synth VRA8」シリーズ第6弾。8ビットCPUひとつで音を合成するモノフォニック（単音）シンセサイザー（MIDI音源）です。スケッチはフリーで公開しており、Arduino Unoと抵抗、コンデンサ、オーディオジャックだけで製作できます。本作品から、サンプリング周波数が31.25 kHzに「倍増」し、シリーズ最高の音質になっています。

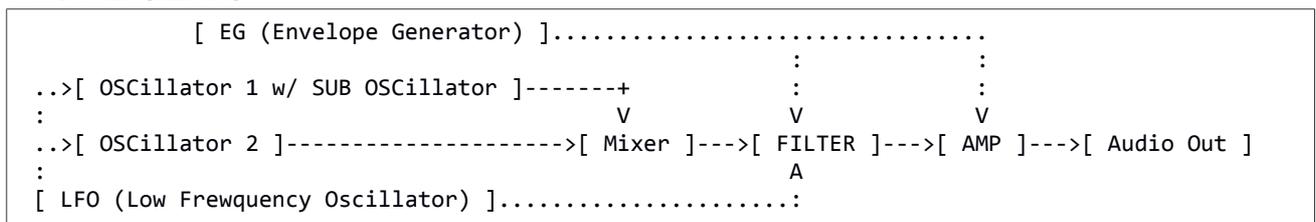
## 1 Digital Synth VRA8-N仕様

- Arduino Uno（8ビットCPU）だけでサウンド生成する、モノフォニック・シンセサイザー（MIDI音源）
- MIDI入力：USBシリアル（38.4 kbps）またはMIDI（31.25 kbps） ※USB MIDI対応も可能
- オーディオ出力：単極PWM（62.5 kHz）+RC回路（カットオフ周波数：15.9 kHz, R：100 Ω, C：100 nF）
- オシレーター波形：のこぎり波・矩形波+サブ正弦波 ● サンプリング周波数/ビット深度：31.25 kHz/8 bit
- 位相分解能：24 bit ● 音域：C-1～C9（中央ドをC4とする） ● ピッチベンド、ポルタメント：対応
- ソースはフリー ● 作り方：<http://www.instructables.com/id/Digital-Synth-VRA8-N-for-Arduino-Uno/>

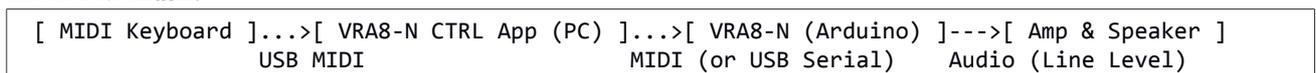
### パラメーター一覧 (v1.0.0)

<b>CUTOFF</b>	<b>RESONANCE</b>	<b>CUTOFF &lt; EG (-/+)</b>	<b>DECAY</b>
<b>OSC2 COARSE (-/+)</b>	<b>OSC2 FINE (-/+)</b>	<b>PORTAMENTO</b>	<b>ATTACK</b>
<b>OSC WAVE (SAW/SQ)</b>	<b>OSC2 MIX</b>	<b>SUB OSC MIX</b>	<b>SUSTAIN</b>
<b>(RESERVED)</b>	<b>(RESERVED)</b>	<b>LEGATO (OFF/ON)</b>	<b>AMP (GATE+RLS/EG)</b>
<b>LFO RATE</b>	<b>LFO DEPTH</b>	<b>LFO &gt; PITCH (2/1+2)</b>	<b>LFO &gt; CUTOFF (-/+)</b>
<b>P. BEND RANGE</b>	<b>(RESERVED)</b>	<b>KEY ASGN (LO/LAST)</b>	<b>(RESERVED)</b>

### ブロック図 (v1.0.0)



### 展示システム構成



## 2 Digital Synth VRA8 シリーズ開発の経緯

作者は、Maker Faire Tokyo 2013を見学後、2014年に当時の勤務先で「ものづくり」の部活動に参加したことを契機にArduinoに触れるようになった。当初は明確に作りたい物が決まっていた訳ではなかったが、過去のDTM経験や「音楽のまち」浜松市に住んでいること等から、シンセサイザー（音源モジュール）を作ることにした。

開発は、実験が行いやすいように Ruby 言語を用いて PC 上でプロトタイプを開発し、それを Arduino に移植するという手順を進めた。同時に、電子楽器に関する技術や文化的背景についての学習を進めた。そして、完成したのが最初の作品「Digital Synth VRA8」である。（詳細については、浜松 Ruby 会議 01 の発表資料「Ruby x Arduino でシンセサイザーを作ってみた」 <https://gist.github.com/risgk/0db52ea683530652d933> を参照）

その後、Web Audio API による作品制作を挟みつつ、Arduino を用いた作品制作を続けてきた。本日展示している「Digital Synth VRA8-N」はシリーズ 6 番目の作品である。モノフォニック（単音）・シンセサイザーではあるものの、サンプリング周波数が 31.25 kHz に「倍増」し、シリーズ最高の音質となっている。

### 3 8 ビット CPU を用いたシンセサイザーの紹介

Arduino や Atmel AVR、その他の 8 ビット CPU を用いた楽器やシンセサイザー作品は数多く存在する。Arduino 環境での楽器の改造や自作、8 ビット CPU の処理能力、RAM、ROM 等の制約、PWM 出力のサウンドに対して、技術的あるいは音楽的なおもしろさを見出す制作者が多いのではないかと考える。一部の作品を以下に紹介する。

- **Audino** Tinker.it の Peter Knight さんの作品。
- **CAmiDion** @きよし (Akiyoshi) さんの作品。
- **Mozzi** Tim Barrass さんのライブラリ。
- **fraAngelico** STANDUINO (現 Bastl Instruments) 社の製品。
- **Atmegatron** Soulsby Synthesizers 社の製品。
- **8bit CPU Synth** akira matsui さんの作品。8051 CPU。
- **ARM Theremin** Haruo Yamashita さんの作品。

### 4 Digital Synth VRA8-N 設計ノート

設計のポイントについて説明する。（前々作「Digital Synth VRA8-P」については [http://risgk.github.io/digital-synth-vra8-p/mft2016\\_isgk\\_instruments.pdf](http://risgk.github.io/digital-synth-vra8-p/mft2016_isgk_instruments.pdf) を、前作「Digital Synth VRA8-Px」については [http://risgk.github.io/digital-synth-vra8-px/mft2017\\_isgk\\_instruments.pdf](http://risgk.github.io/digital-synth-vra8-px/mft2017_isgk_instruments.pdf) を参照）

- **\_\_attribute\_\_((always\_inline))** Arduino IDE は GCC を使用しているが、最適化オプションが -Os (バイナリサイズ最適化) 固定である。そのため、この属性を指定することによって、全関数を強制的にインライン展開している。
- **noInterrupts 関数** Arduino は、標準関数 micros などの実現のために、デフォルトで Timer0 割り込みを使用している。PWM 出力用の Timer0 が相当な CPU 負荷につながっていたが、前作から割り込みを禁止した。
- **DEBUG 用機能** 前作から、DEBUG 時のみタイマーのカウンタ値を MIDI 出力 (シリアル出力) させる機能を追加した。正確な時間計測が可能になったことで、CPU を無駄なく使用できるようになった。（なお、MIDI 受信による発音開始タイミング等にサウンド生成の計算が追いつかなくなる場合があるが、聴感上問題がないと判断している）
- **パラメーターの選択** 市販されている小型フィジカル・コントローラー (MIDI キーボード等) でも音作りを楽しめるよう、パラメーターを厳選している。開発で一番時間をかけているのは、パラメータの仕様を考えることかもしれない。今作では 16 個を超えるパラメーターを用意してしまったが、果たして使いやすくできているだろうか？
- **「ゆらぎ」** 本作はデジタル・シンセだが、前作に続いて、アナログ・シンセらしい「ゆらぎ」の導入を試している。具体的には、各オシレーターのパッチを「レッド・ノイズ」のサンプル&ホールドで変化させている。
- **低音優先のキー・アサイン** デフォルト設定では、複数のキーを押したときに一番低いキー (ノート) のピッチで発音される。また、シングル・トリガーになる。これにより、モノフォニック・シンセサイザー独特の演奏方法が可能になる。前作のように、後着優先 (最後のキーしか記憶しない仕様)、マルチ・トリガーにすることもできる。
- **指数関数的ピッチ変化のポルタメント** 前作までは、ピッチを linear に (定速で) 変化させてきたが、今作では exponential に (定時間で) 変化。レガート演奏時のみ自動でポルタメントがかかるオプションも用意。
- **ソースコード** 仕様・設計変更整理整頓が追いつかず、色々とゴミが残っている……。 (すみません)