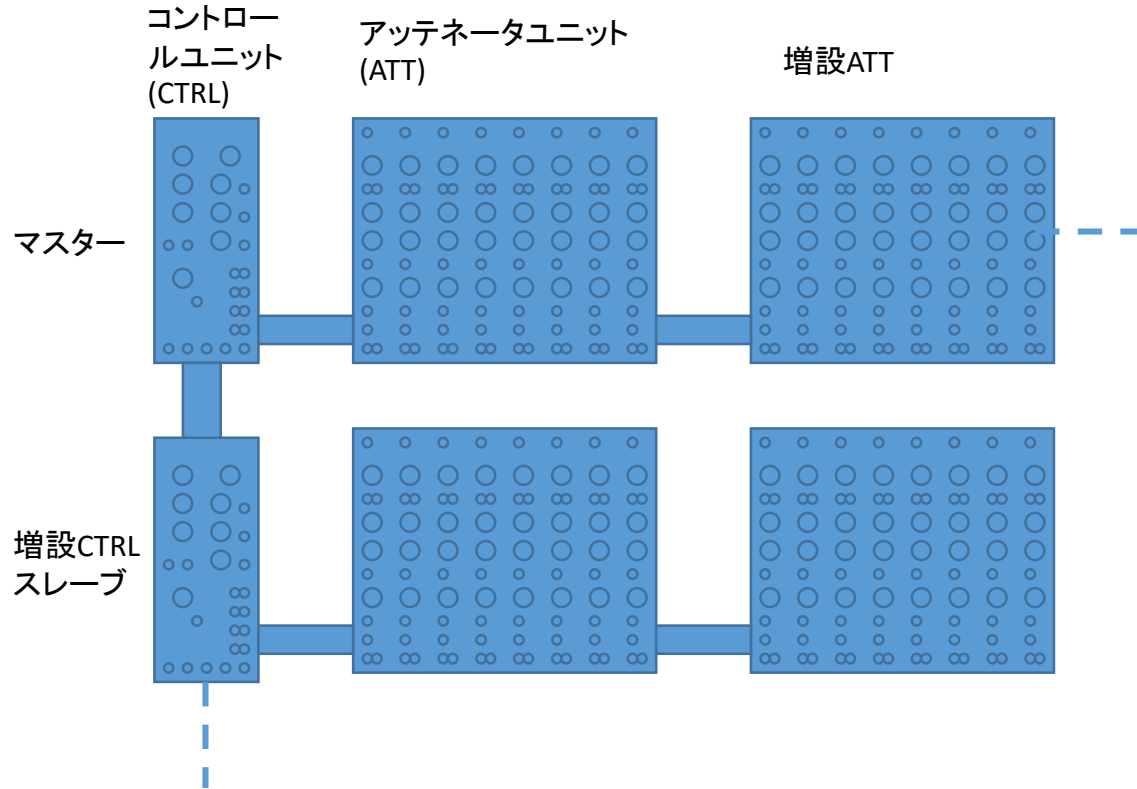


安西さんのシーケンサーまとめ

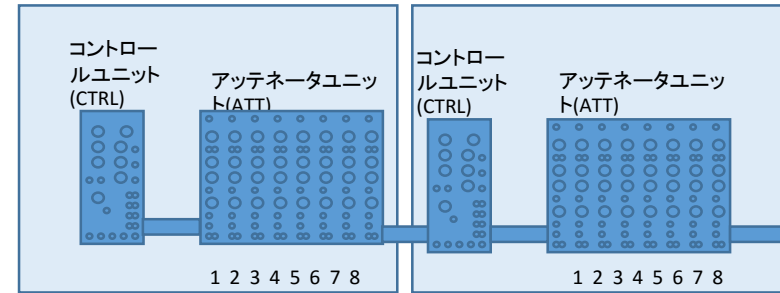
2015/12/25版

# 基本コンセプト

- シーケンサーのコントローラ部(CTRL)とアッテネータ部(ATT)を分割
- ユニット間はフラットケーブルで接続
- 1つのATTは3段x8ステップ
- ATTを横方向に増設してステップ数を増やす事ができる(最大64ステップ)
- CTRLを縦方向に増設して段数を増やす事ができる。この時、独立、クロックのみ同期、完全同期などの動作モードが選択できる
- 横方向にCTRLを挟んで増設した場合、それぞれ独立させたり連続させたり(SeriesFromLeft)できる
- ステップを分割するラチェットコントローラ(オプション)

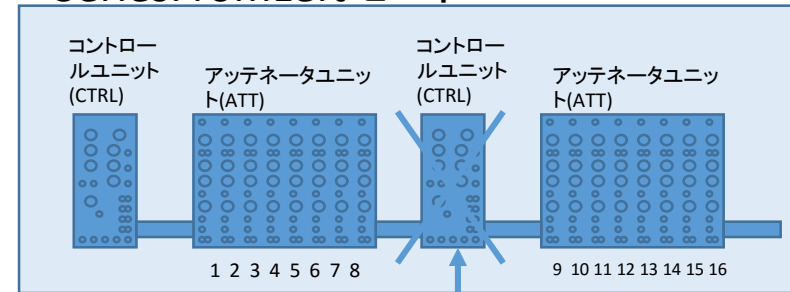


## 通常モード



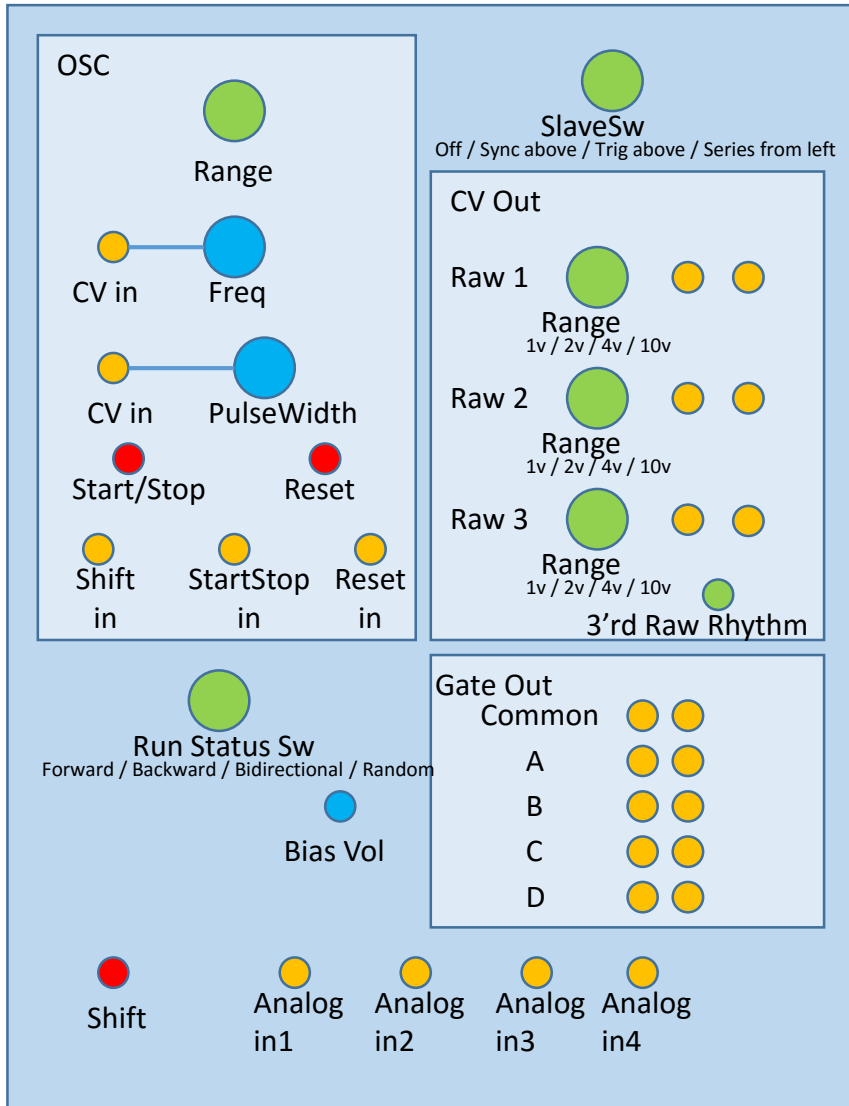
8ステップ  
が2台

## SeriesFromLeftモード





16ステップ  
が1台

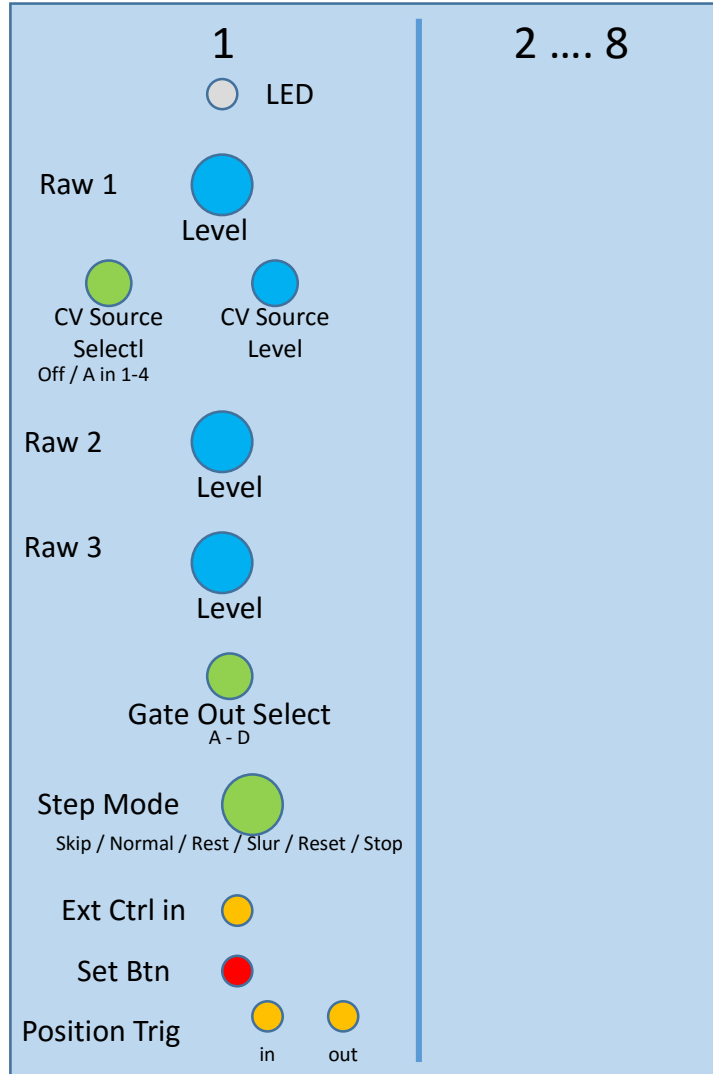
# CTRL 詳細



- Range シーケンスの速さのレンジ切り替え
- Freq シーケンスの速さ、CVでの制御も可
- PulseWidth Gate信号の長さを指定、CVでの制御も可
- Start/Stop シーケンサーのスタート/ストップ
- Reset (スキップでない)最初のステップに戻る
- Shift 強制的に1ステップ進める
- Shift in 外部信号でShiftボタンと同じ動作
- StartStop in 外部信号でStart/Stopボタンと同じ動作
- Reset in 外部信号でResetボタンと同じ動作
- SlaveSw off - 通常動作  
SyncAbove - マスターのシーケンサと完全に同期  
TrigAbove - マスターと同じトリガーで動作(ステップの進み方は独立)  
SeriesFromLeft - このCTRLは無視され、右に接続されているATTは左のATTのシーケンスの続きになる
- Raw 1 Raw1の出力。Rangeで出力レンジを切り替え
- Raw 2 Raw2の出力。Rangeで出力レンジを切り替え
- Raw 3 Raw3の出力。Rangeで出力レンジを切り替え
- 3'rdRawRhythm Raw3の出力でステップの進む速度を設定
- RunStatusSw Forward - 通常動作  
Backward - ステップを逆方向に進める  
Bidirectional - ステップを往復して進める  
Random - ステップをランダムに進める。BiasVolで動作が変化
- Bias Vol Random時の乱数の幅の設定。最小時は演奏可能な最初のステップのみ選択され、大きくすると乱数の幅(選択対象のステップ)が増加してゆく。最大で演奏可能な全てのステップが対象となる
- GateOut A-D ゲート出力A-D。ATT側のGateOutSelectで出力される端子を指定
- GateOut Common 共通ゲート出力。ゲート出力A-DのOR
- Analog in 1-4 ATT側のCVSourceSelectがOff以外の時、ここに入力されたCVを元にCVが出力される

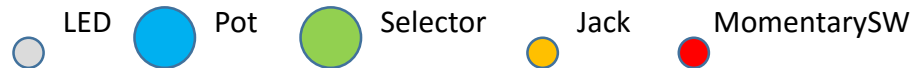
-  Pot
-  Selector/ToggleSW
-  Jack
-  MomentarySW

# ATT 詳細



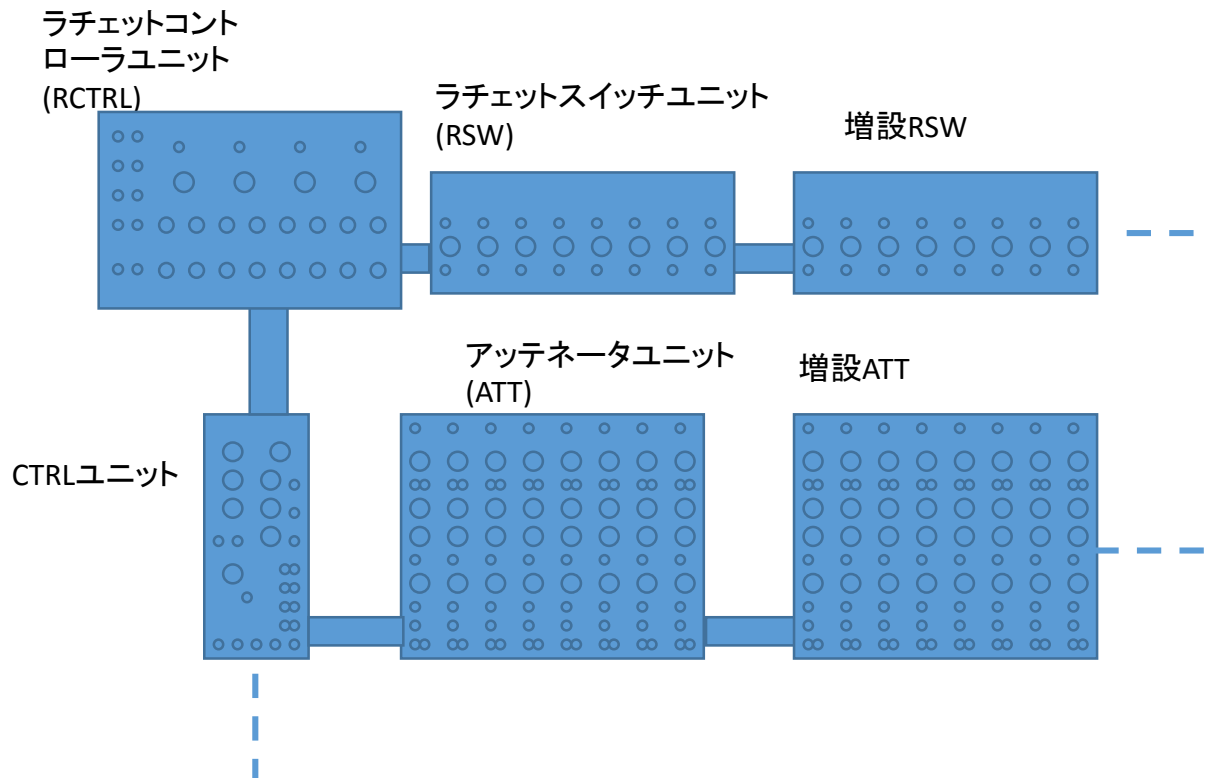
以下の説明は1ステップ分であり、同じものが8ステップ分並んでいる

LED	シーケンスがこのステップに来たら点灯
Raw1Level	このステップでのRaw1のCV出力レベルの設定
CV Source Select	off 何もしない。CV Source Levelも無効 A in 1-4 CTLのAnalog in 1-4 の信号がCV Source Levelで調整されてRaw1のCVに加算される
CV Source Level	CV Source Levelで選択されたAnalog in の信号を加算するレベルの調整
Raw2Level	このステップでのRaw2のCV出力レベルの設定
Raw3Level	このステップでのRaw3のCV出力レベルの設定
GateOutSelect	このステップでのGate信号を出力する端子の設定。 Gate Out Commonにはこの設定には関わらず出力される
StepMode	このステップの動作モードの設定 Skip - このステップには止まらず次に進む Normal - 通常動作。設定されたCVとGateが出力される Rest - CVは直前の値をホールドし、Gateを出力しない Slur - CVは通常通り出力、Gateは出力しっぱなしになる Reset - 演奏可能な一番若いステップに戻る Stop - このステップでシーケンサーを停止
Ext Ctrl in	CV入力によってStepModeが設定される。この端子を使用する場合はStepModeノブは無効となる。StepModeと選択肢の順序が違うのに注意 (0V..Normal / 1V..Skip / 2V..Rest / 3V..Slur / 4V..Reset / 5V..Stop)
Set Btn	このボタンを押すと強制的にこのステップに移動
Position Trig in	Gate信号が入力されると強制的にこのステップに移動。適当なPosition Trig out と Position Trig in 間を接続する事でループやスキップを行う事ができる
Position Trig out	シーケンスがこのステップに来たら出力されるGate信号。

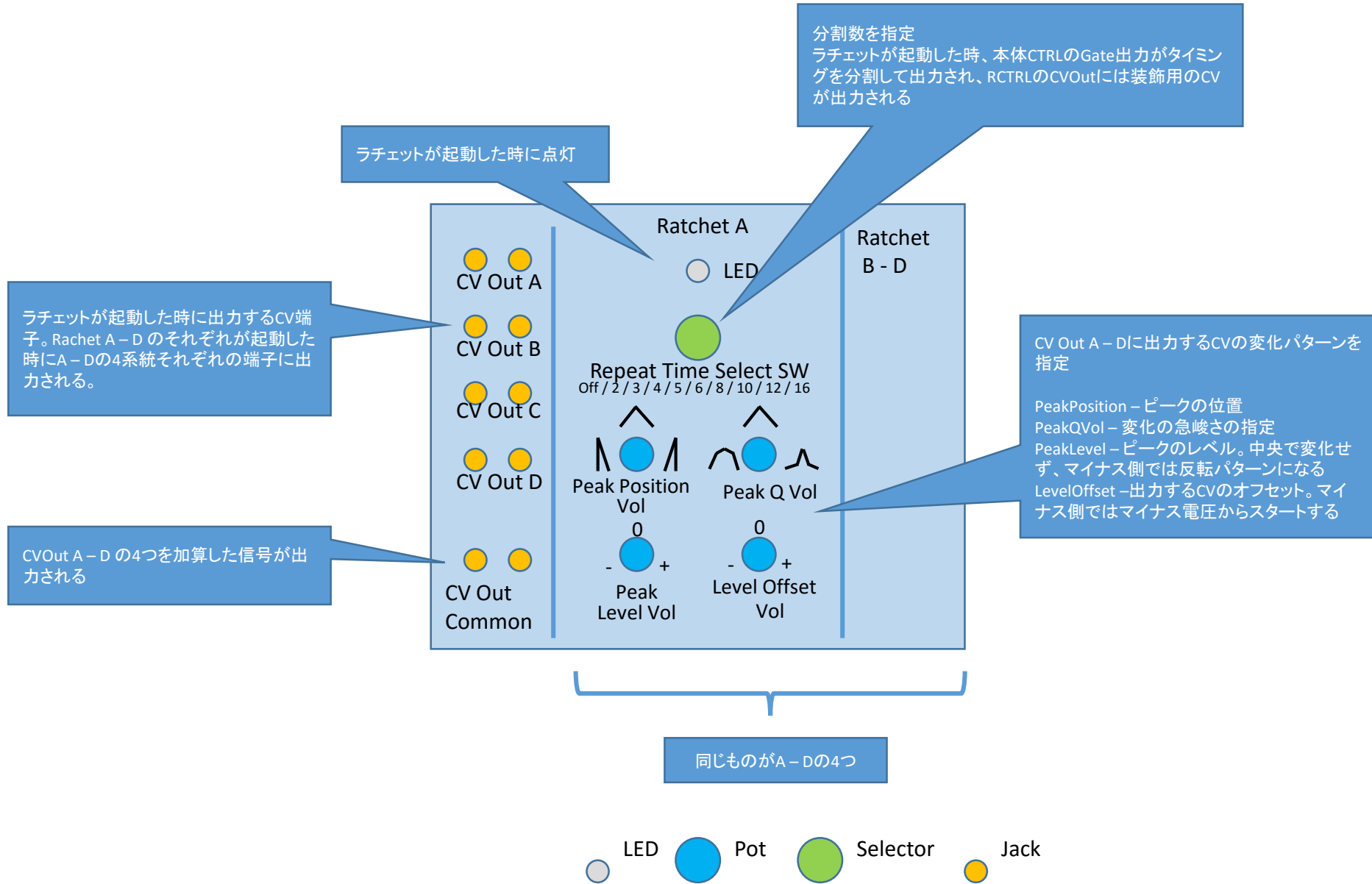


## ラチェットコントローラ

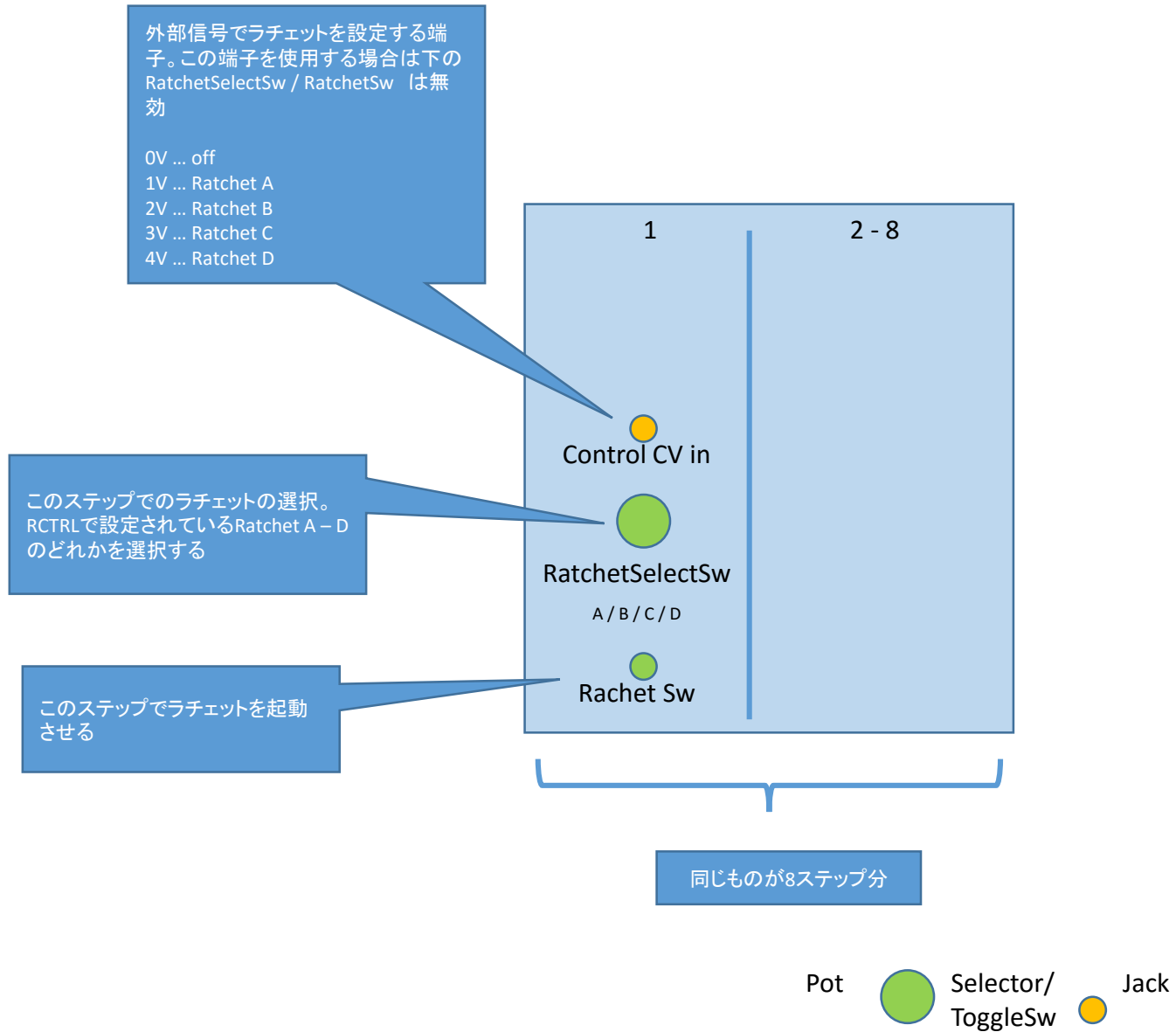
- シーケンサーの1ステップをさらに細かく分割するオプションユニット
- ラチェットコントローラユニット(RCTRL)とラチェットスイッチユニット(RSW)から成る
- RSWの各ステップは下にあるATTの1ステップに対応し、ATTと同様に横に接続して増設できる
- それぞれのステップで分割が指定されていればそのステップでラチェット機能が働く
- 分割数は、Off / 2 / 3 / 4 / 5 / 6 / 8 / 10 / 12 / 16から選択
- ラチェットを指定するとそのステップでの本体CTRLのGate出力は自動的に細かい連続パルスに分割される
- 装飾用のCVが本体CTRLとは別途にRCTRLより出力される



# ラチェットコントローラユニット(RCTRL) 詳細



# ラチェットスイッチユニット(RSW) 詳細

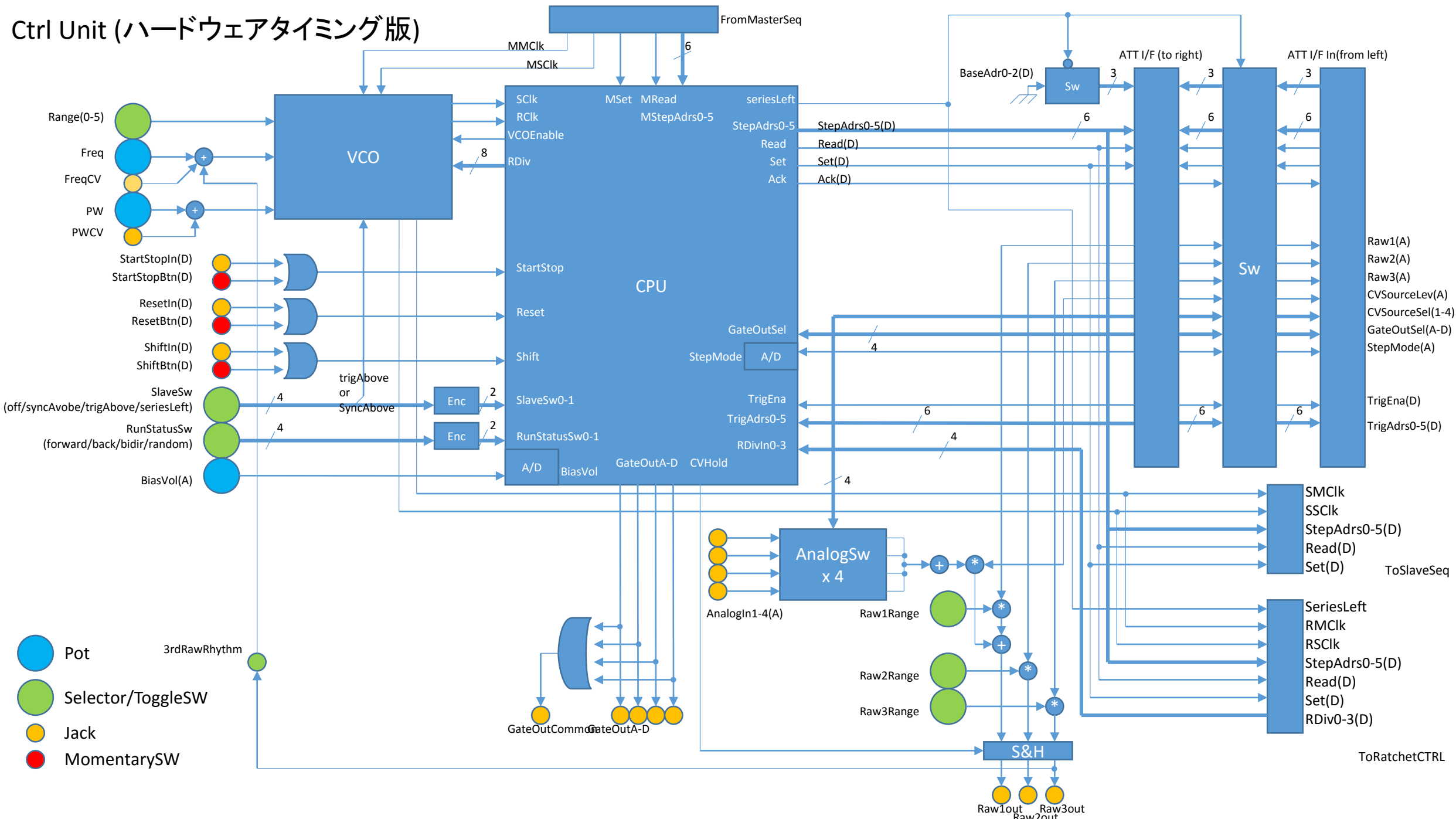


## 以降は内部構成案

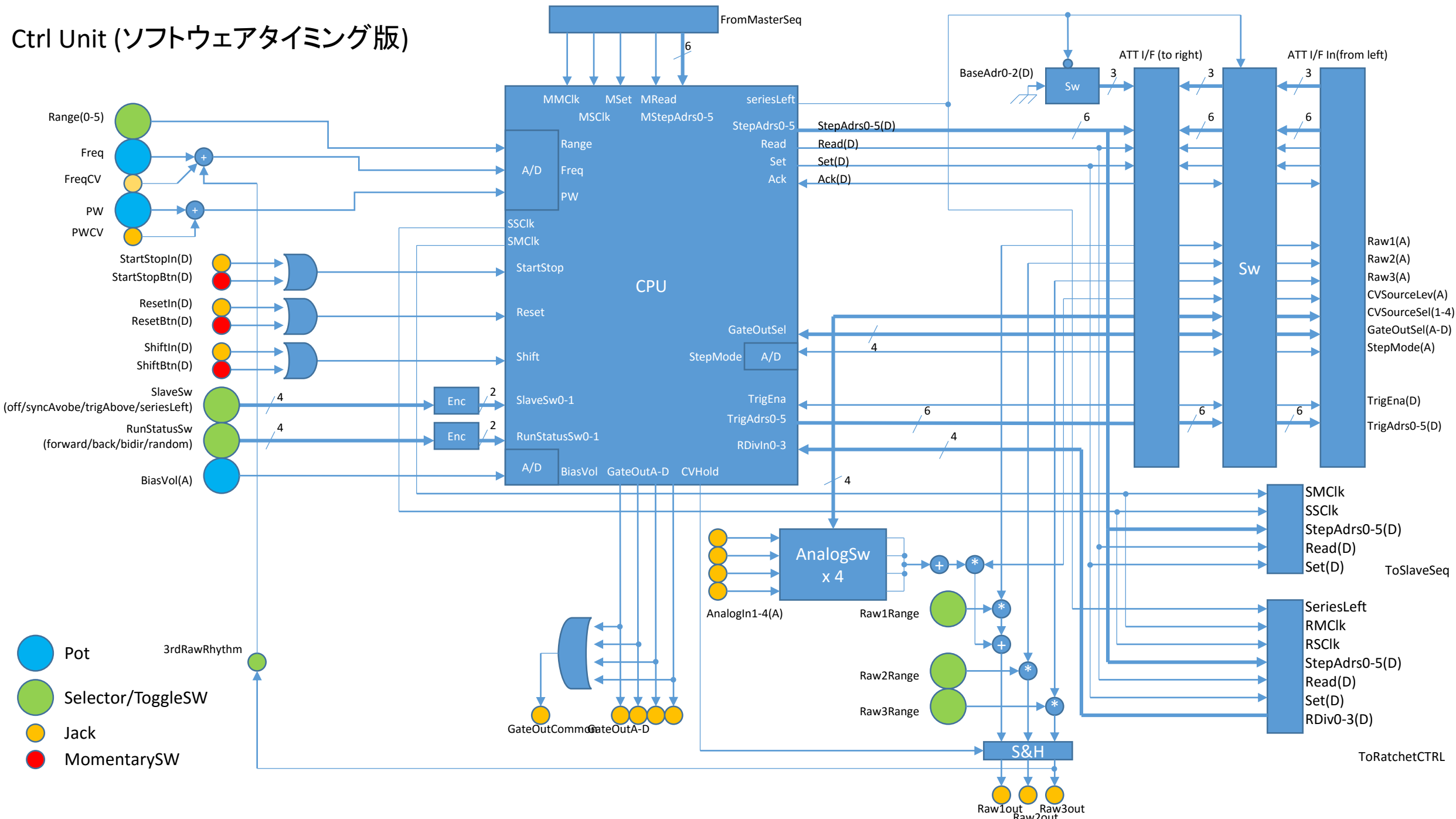
- CPUとしてArduino MEGA 2560を使用する事を想定
- CTRLユニットの構成はハードウェアのオシレータで動作タイミングを取るハードウェアタイミング版とソフトウェアでタイミングを取るソフトウェアタイミング版の2案



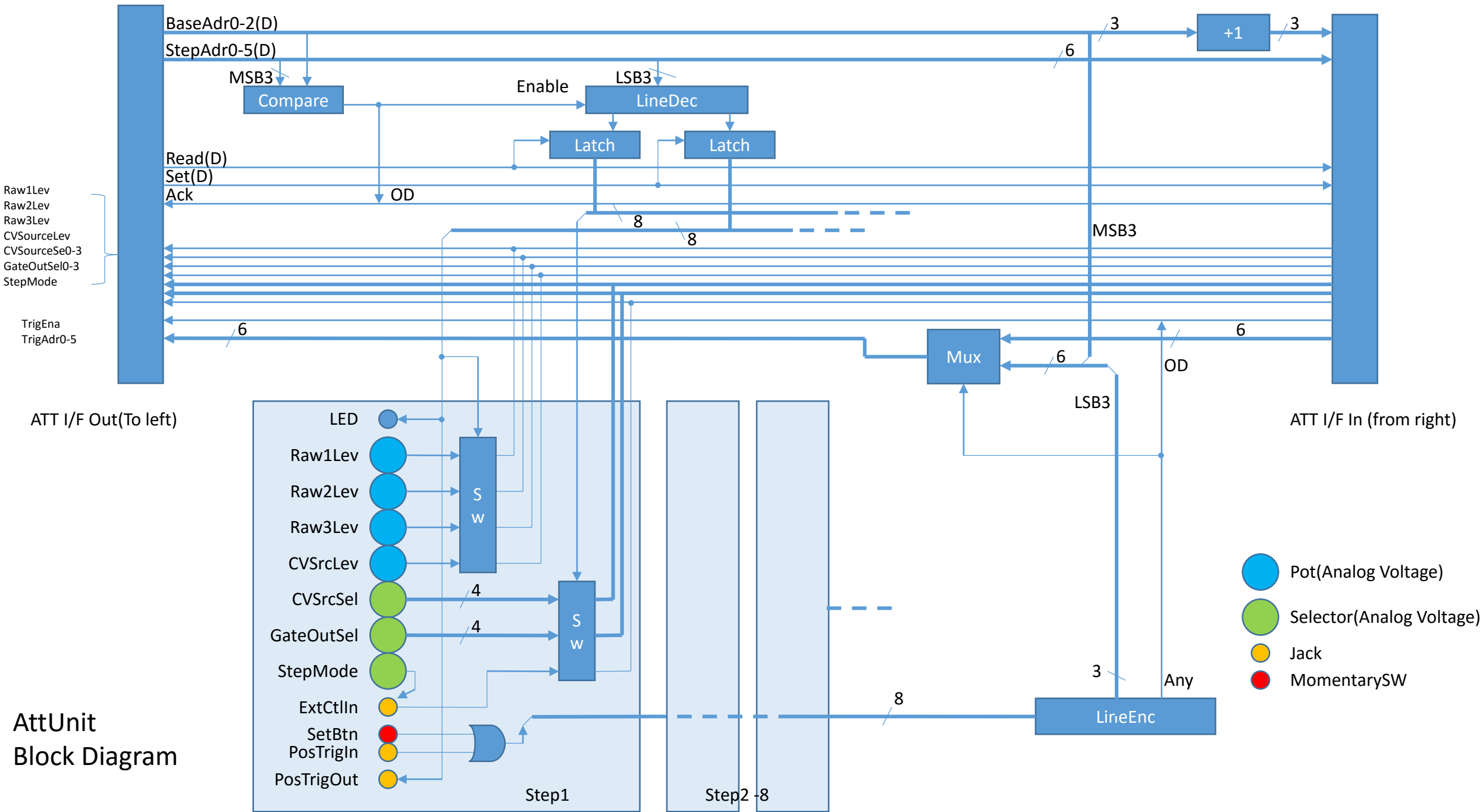
# Ctrl Unit (ハードウェアタイミング版)

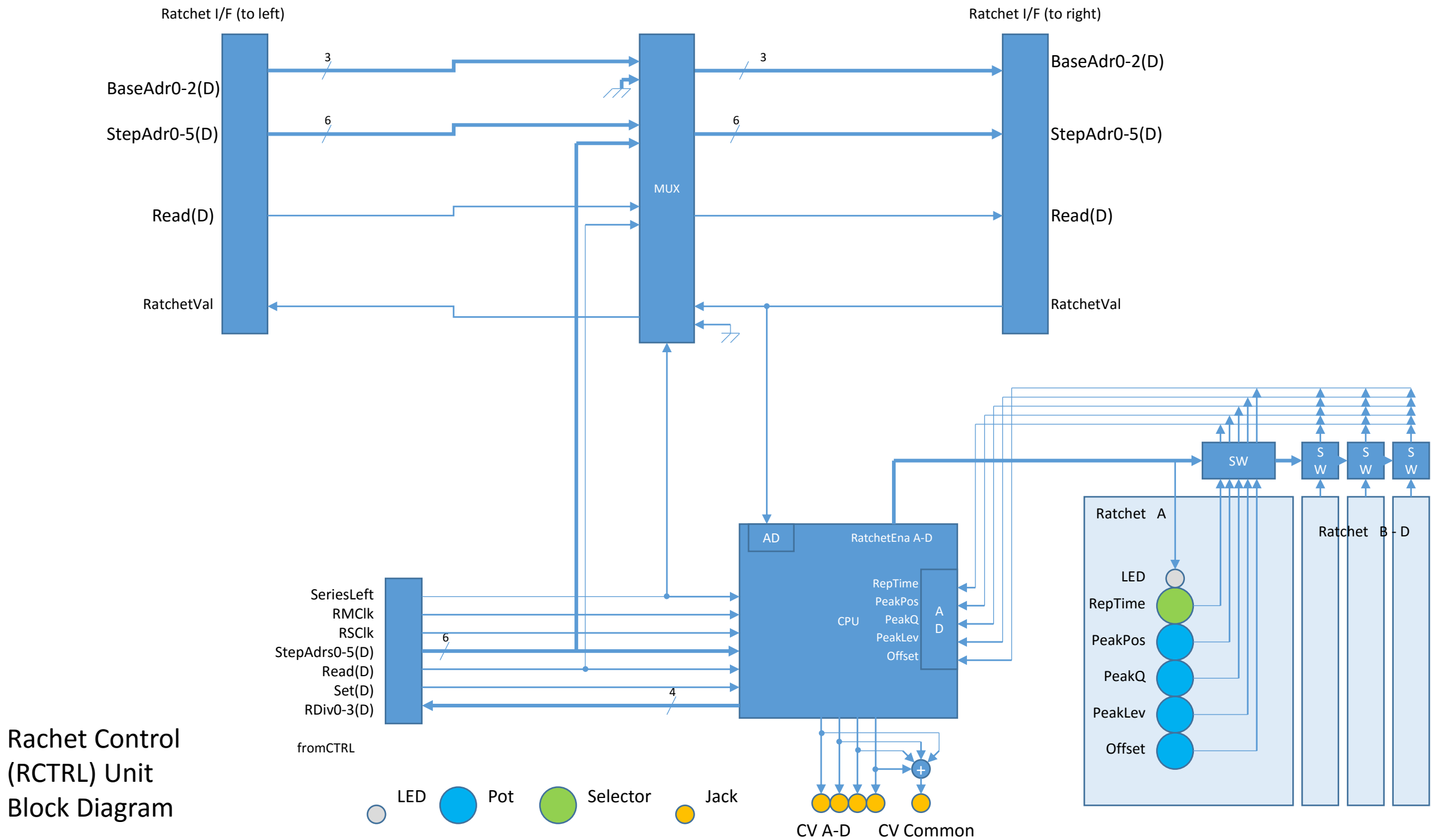


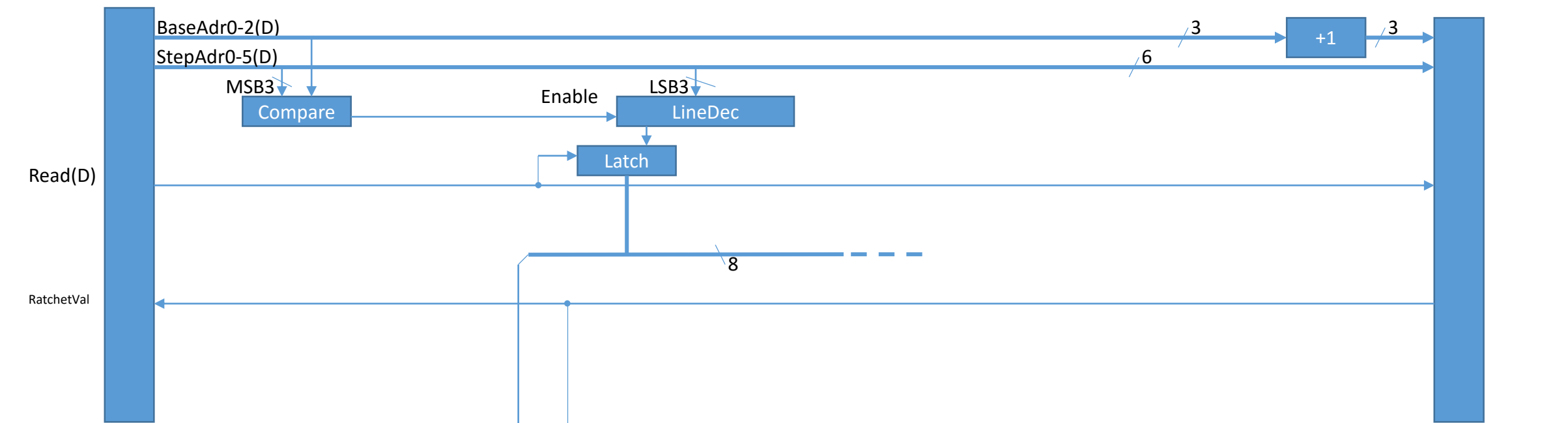
# Ctrl Unit (ソフトウェアタイミング版)



- Pot
- Selector/ToggleSW
- Jack
- MomentarySW

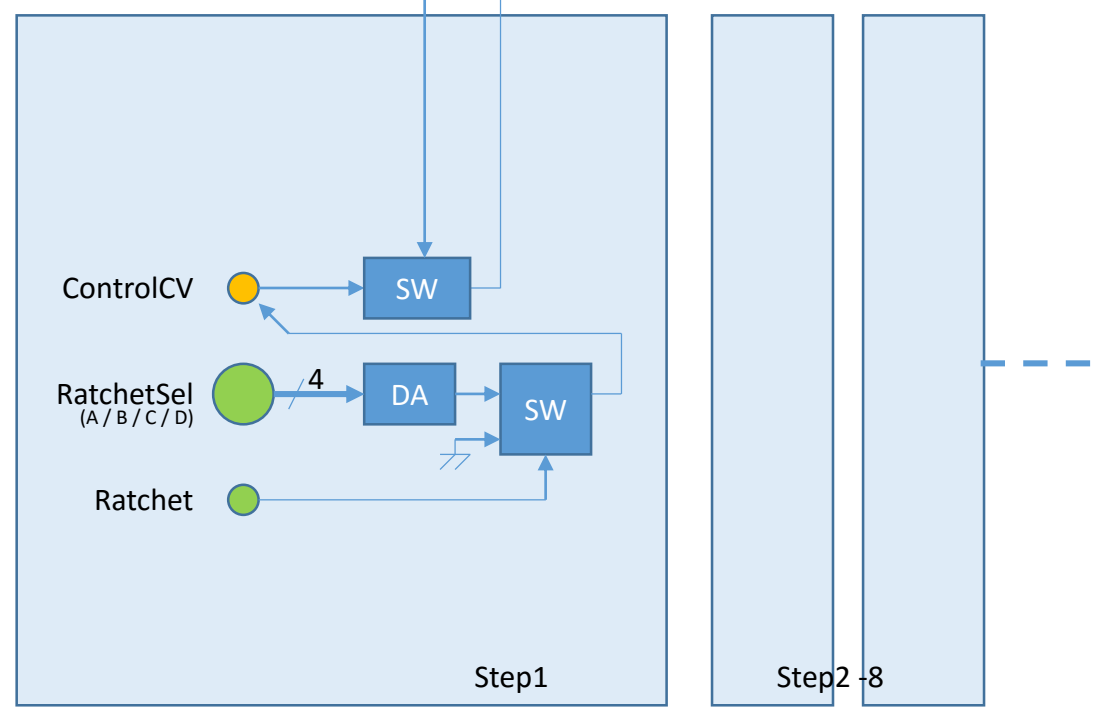






Ratchet I/F Out (To left)

Ratchet I/F In (from right)



- Selector/ToggleSw
- Jack

Ratchet Switch (RSW) Unit Block Diagram

# CPU部 Arduino Mega へのピン割り当て

## ハードウェアタイミング版

種別	名前	本数	Digital/ Analog	I/O	Arduino	説明
Clock (ハードウェア タイミング版)	SClk	1	D	I	2 (int 0)	ステップクロック
	RClk	1	D	I	18 (int 5)	ラチェットクロック
	VCOEnable	1	D	O	29	VCOイネーブル
	RDiv	8	D	O	A8 - A15	ラチェットクロック分割

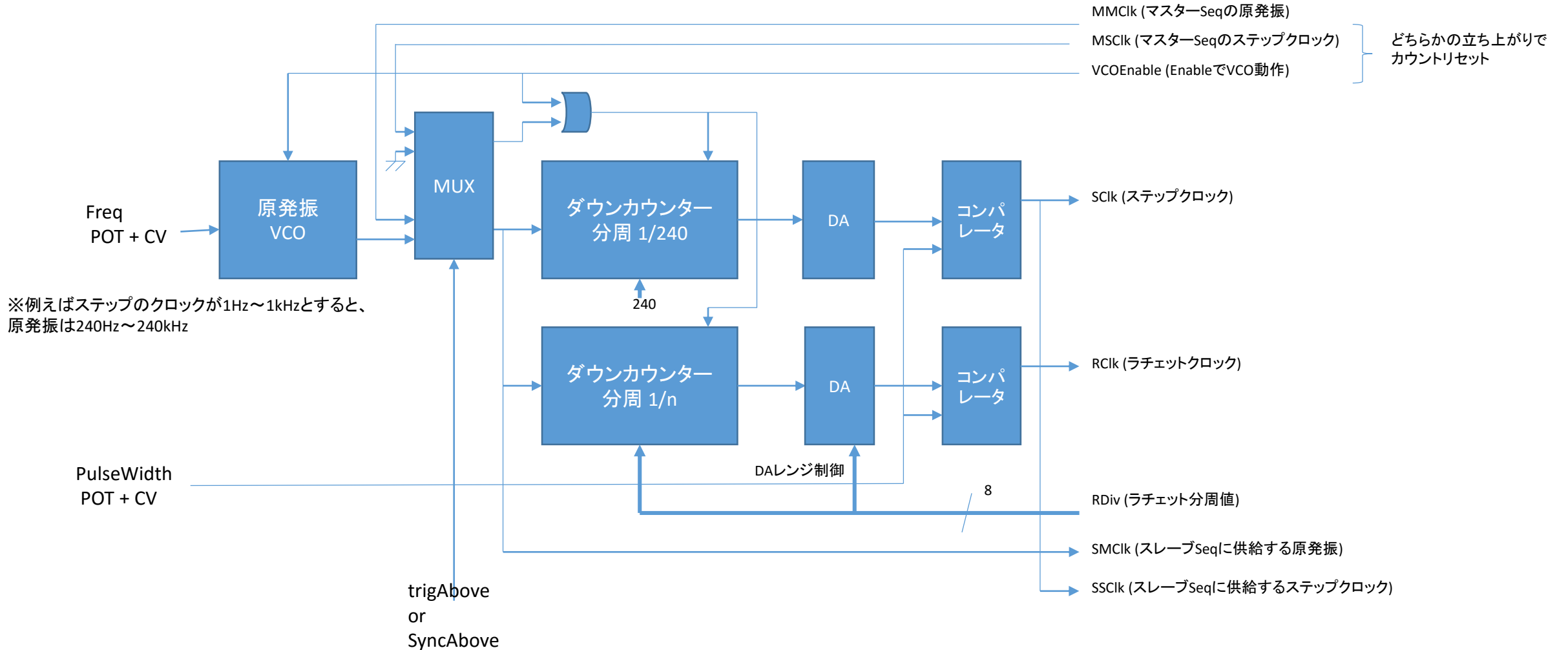
## ソフトウェアタイミング版

種別	名前	本数	Digital/ Analog	I/O	Arduino	説明
Clock (ソフトウェア タイミング版)	MMClk	1	D	I	2 (int 0)	マスターのメインクロック
	MSClk	1	D	I	18 (int 5)	マスターのステップクロック
	SMClk	1	D	O	29	スレーブのメインクロック
	SSClk	1	D	O	30	スレーブのステップクロック
	Range	1	A	I	A2	VCOレンジ
	Freq	1	A	I	A3	VCO周波数
	PW	1	A	I	A4	VCO Pulse Width

種別	名前	本数	Digital/ Analog	I/O	Arduino	説明
User Control	StartStop	1	D	I	3 (int1)	シーケンサーのスタートストップ
	Reset	1	D	I	21 (int2)	最初のステップに戻る
	Shift	1	D	I	20 (int3)	強制的に1ステップ進める
	SlaveSw	2	D	I	32 - 33	スレーブモード (off/syncAbove/trigAbove/SeriesLeft)
	RunStatus	2	D	I	34 - 35	ステップの進み方の制御 (forward/backward/bidir/random)
	BiasVol	1	A	I	A1	randomモードでの乱数の幅
ATT I/F	SeriesLeft	1	D	O	36	左右のATTを接続する(SeriesLeftモード)
	StepAdrs	6	D	O	4 - 9	CTRLユニットが処理しようとするステップ
	Read	1	D	O	10	StepAdrsが示すステップの設定値要求
	Set	1	D	O	13	StepAdrsが示すステップに移動
	Ack	1	D	I	12	StepAdrsに対するATTユニットの応答
	GateOutSel	4	D	I	38 - 41	出力するGate端子の選択
	StepMode	1	A	I	A0	このステップのモード (Skip/Normal/Rest/Slur/Reset/Stop)
	TrigEna	1	D	I	19 (int4)	PositionTrig/Setボタン入力
	TrigAdrs	6	D	I	42 - 47	PositionTrig/Setボタン入力のStepAdrs
Gate Output	CVHold	1	D	O	48	CVを一時的にホールドする
	GateOut	4	D	O	50 - 53	ゲート出力
	RDivIn	4	D	I	14 - 17	ラチェット分割指示
	MSet	1	D	I	11	マスターのSet信号
	MRead	1	D	I	22	マスターのRead信号
	MStepAdrs	6	D	I	23 - 28	マスターのStepAdrs

# ハードウェアタイミング版VCO部の構成

- VCOはステップのクロックの240倍の周波数を原発振として、分周によりラチェットクロックとステップクロックを発生する
- ステップクロックはシーケンサの1ステップのクロック
- ラチェットクロックは現在のステップのラチェットの分割数倍のクロック
- ラチェットクロックの分周値は  $240/\text{分割数}$ 、= 240/120/80/60/48/40/30/24/20/15から選択
- PulseWidthはそれぞれのクロックに対しDuty比で制御する



## ATT I/F Signal map

BaseAdrs0-2	Digital	CTRL=>ATT	ATTユニットの接続された位置。StepAdrsの上位3bitに相当
StepAdrs0-5	Digital	CTRL=>ATT	CTRLユニットが処理しようとするステップ(0-63)
Read	Digital	CTRL=>ATT	StepAdrsで示すステップの設定値(StepModeとGateOutSel)を要求する。 該当するATTはこれによりCTRLにAckとStepMode、GateOutSel0-3の設定値をCTRLに返す
Set	Digital	CTRL=>ATT	StepAdrsに該当するATTはそのステップのLEDを点灯、PositionTrigを出力。 またRaw1Lev、CVSrcSel0-3、CVSrcLev、Raw2Lev、Raw3Levの設定値をCTRLに返す CTRLは返された設定値からRaw1-3のCVを出力 StepAdrsに該当するATTはこの信号で応答する
Ack	Digital	CTRL<=ATT(OD)	
Raw1Lev	Analog	CTRL<=ATT	Raw1Levの設定値。CTRLはこの電圧を元にRaw1のCVを出力する。 (Raw1CV = Raw1Lev * Raw1Range + CVSrc * CVSrcLev)
CVSrcSel0-3	Digital	CTRL<=ATT	CVSrcSelの設定値。CTRLはこれで選択されたAnalogInの信号をRaw1CVに加算
CVSrcLev	Analog	CTRL<=ATT	CVSrcLevの設定値。CVSrcSelで選択されたAnalogInの信号を加算するレベル
Raw2Lev	Analog	CTRL<=ATT	Raw2Levの設定値。CTRLはこの電圧を元にRaw2のCVを出力する (Raw2CV = Raw2Lev * Raw2Range)
Raw3Lev	Analog	CTRL<=ATT	Raw3Levの設定値。CTRLはこの電圧を元にRaw3のCVを出力する (Raw3CV = Raw3Lev * Raw3Range)
GateOutSel0-3	Digital	CTRL<=ATT	GateOutSelの設定値。CTRLのCPUで処理される
StepMode	Analog	CTRL<=ATT	StepModeの設定値。CTRLのCPUで処理される
TrigEna	Digital	CTRL<=ATT(OD)	PosTrigInまたはSetボタンのどれかが入力された事を通知
TrigAdrs	Digital	CTRL<=ATT	入力されたPosTrigInまたはSetボタンのステップ(0-63)



## CPU部 Signal map

	タイプ	入出力	本数	説明
クロック/マスター関係				
SClk	Digital	In	1	シーケンサーのステップクロック
RClk	Digital	Out	1	ラチェットクロック。現在のラチェットの倍数 x ステップクロック。起動していない時はClkと同じ
MSet	Digital	In	1	マスターからのSet信号
MRead	Digital	In	1	マスターからのRead信号
MStepAdrs0-5	Digital	In	6	マスターからステップアドレス
VCOEnable	Digital	Out	1	VCOの動作の開始/停止
RDiv0-7	Digital	Out	8	現在のラチェットの分周比 =240/分割比 offの時は240
RDivIn0-4	Digital	In	4	ラチェットコントローラから受け取る現在のラチェットの分割比 (off/2/3/4/5/6/8/10/12/16のインデックス)
ユーザー制御関係				
StartStop	Digital	In	1	シーケンサーのスタートストップ
Reset	Digital	In	1	最初のステップに戻す
Shift	Digital	In	1	強制的に1ステップ進める
SlaveSw0-1	Digital	In	2	スレーブモード(off/syncAbove/trigAbove/SeriesLeft)
RunStatusSw0-1	Digital	In	2	ステップの進み方の制御(forward/backward/bidir/random)
BiasVol	Analog	In	1	Randomモードでの乱数の幅
ATT インターフェース				
SeriesLeft	Digital	Out	1	左右のATTを接続する。SeriesLeftモード時に使用
StepAdrs0-5	Digital	Out	6	CTRLユニットが処理しようとするステップ(0-63)
Read	Digital	Out	1	StepAdrsで示すステップの設定値(StepModeとGateOutSel)を要求する
Set	Digital	Out	1	StepAdrsで示すステップにシーケンスを移動する
Ack	Digital	In	1	StepAdrsに該当するATTはこの信号で応答する
GateOutSel	Digital	In	4	出力するGate端子の選択
StepMode	Analog	In	1	このステップのモード(Skip / Normal / Rest / Slur / Reset / Stop)
TrigEna	Digital	In	1	PositionTrig、Setボタンが入力された
TrigAdrs0-5	Digital	In	6	PositionTrig / Setボタンが入力されたステップ
ゲート出力関係				
CVHold	Digital	Out	1	CVを一時的にホールドする
GateOutA-D	Digital	Out	4	ゲート出力

デジタル出力 24本

デジタル入力 32本

アナログ入力 2本 計58本

# CPU部1ステップの動作 (ラチェット未対応)

SlaveSw=offの場合クロックはClk、 SlaveSw=trig aboveの場合クロックはMClk (Rise Edge)

- 1) CTRLは現在のステップをStepAdrs0-5に出力、Readを出力
- 2) CTRLはAckを確認、応答が無い場合ステップを最初に戻し1)へ
- 3) CTRLはStepModeとGateOutSelを読み取る
- 4) StepMode=Skipの場合、ステップを1つ進め(\*1) 1)へ
- 5) StepMode=Resetの場合、ステップを最初に戻し1)へ
- 6) StepMode=Restの場合、CVHoldにより現在のCVをホールド。Rest以外ならCVHoldをオフ
- 7) CTRLはSetを出力。ATTはこれによりLED点灯、PositionTrigを出力。またRaw1-3Lev、CVSrcSel、CVSrcLevをCTRLに返す。CTRLは返された設定からRaw1-3のCVを出力
- 8) CTRLはStepMode=RestならGateOutA-Dをオフ、そうでなければGateOutSelで指定のポート(GateOut A-D)にゲートを出力
- 9) StepMode=Stopなら停止状態へ
- 10) ステップを1つ進め(\*1)次のRise Edgeを待機

\*1 – 進める順序はCTRLのRunStatusSw(forward / backward / bidirectional / random+BiasVol) に依存  
スキップ等の処理のタイミングが厳しいようならこのロジックとは別にStepModeをポーリングして演奏可能なステップをキャッシュしておくなどが必要かも

## (Falling Edge)

- 1) 処理中のStepModeがSlurでなければGateOutA-Dをオフ

## SlaveSw=SyncAbove

- 1) CTRLは入力されるMStepAdrs0-5を常にStepAdrs0-5へ転送
- 2) MReadが入力された時、自分のReadIに出力し、ステップの設定値(StepMode、GateOutSel)を読み取り
- 3) MSetが入力された時、  
StepModeがRestならCVHoldでRaw1-3のCVをホールド、そうでなければCVHoldをオフ  
自分のSetを出力しStepModeがRestでなければGateOutSelで指定のポート(GateOut A-D)にゲートを出力、RestならGateOutA-Dをオフ
- 4) MClk falling Edgeが入力された時、処理中のStepModeがSlurでなければGateOutA-Dをオフ

## SlaveSw=SeriesFromLeft

- 1) CTRLはSeriesFromLeft信号を出力して左右のATT I/Fを直結、StepAdrsその他の出力を一切しない

## TBD

ラチェット機能とは?

ラチェットコントローラのページ参照

Shift / StartStop / Reset はエッジトリガー?

ExtCtlInによるStepModeの変更は単純に加算? セレクタでの設定が最低値(Skip)以外の場合は?

ExtCtlInを使用する場合はStepModeノブは無効

Analog InおよびCVSrcLevのCV加算はRaw1Rangeの影響を受ける?

StepMode=RestでCVホールド中もAnalog InからのCVは加算される?

SeriesFromLeft状態のCTLからのCV / Gateの出力はどうなるか?

TrigAbove状態のCTRLのStart/Stopは有効? あるいはマスターに影響される?