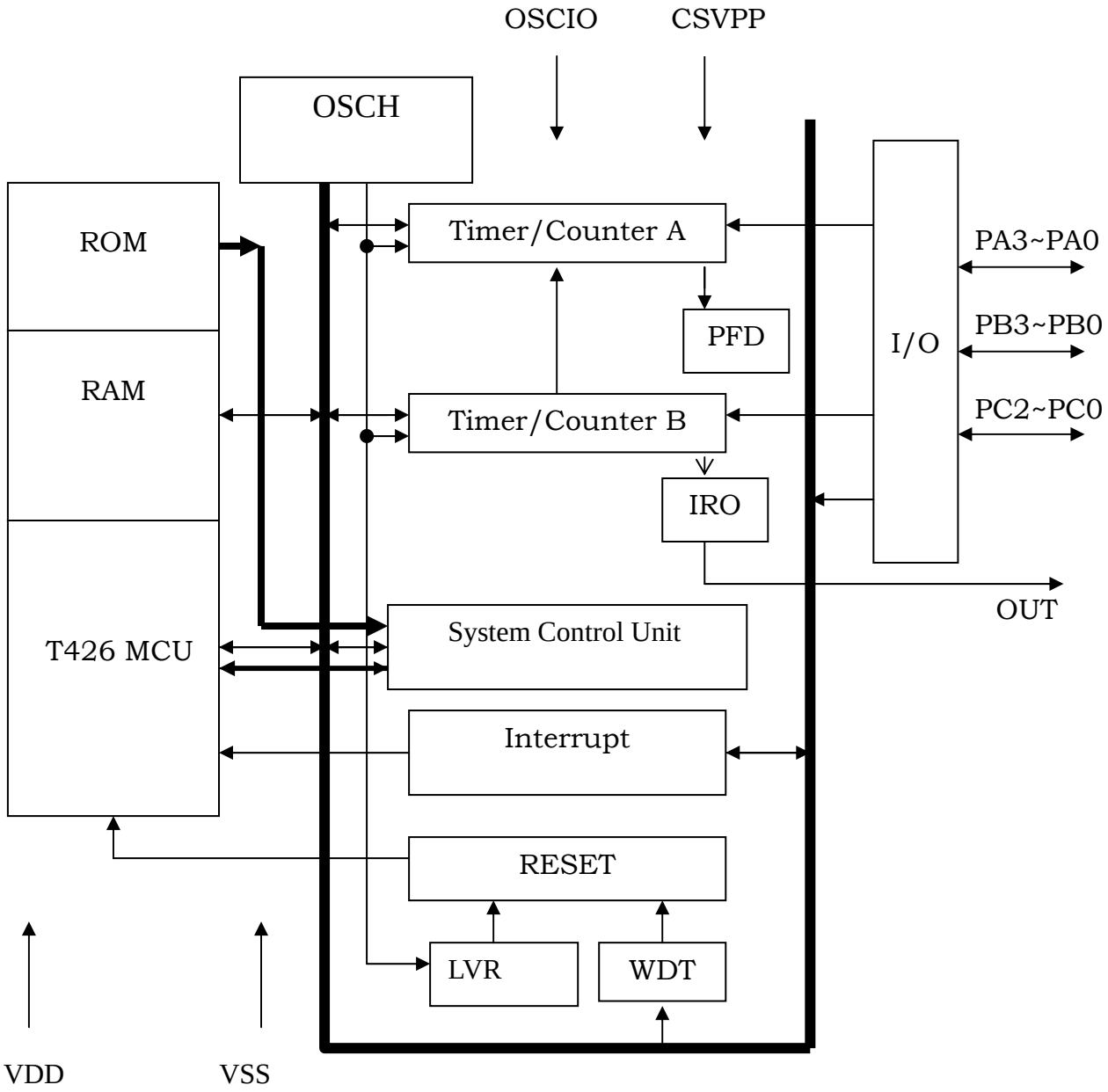
◆ General Description

TTP024 is a high performance remote control transmitter with easy used 4-bits CPU base microcontroller，it is specially designed for use on infrared remote control applications。 All protocols are available by Programming。 The Timing & Format of Bit-A、Bit-B、Bit-C、Bit-D、Bit-0、Bit-1，and theFrame Timing of Sequential Output can be programmable to meet the need of customer。The transmission code consists any type including “ Custom Code(Max. 32 Bits) ” 、 “ Custom Inverse Code(Max. 32 Bits) ” 、 “ Data Code(Max. 8 Bits) ” 、 “ Data Inverse code(Max. 8 Bits) ” 、 “ Bit-A ” 、 “ Bit-B ” 、 “ Bit-C ” 、 “ Bit-D ” 、 “ Bit-0 ” 、 “ Bit-1 ” 、 “ Toggle Bit ” …..。

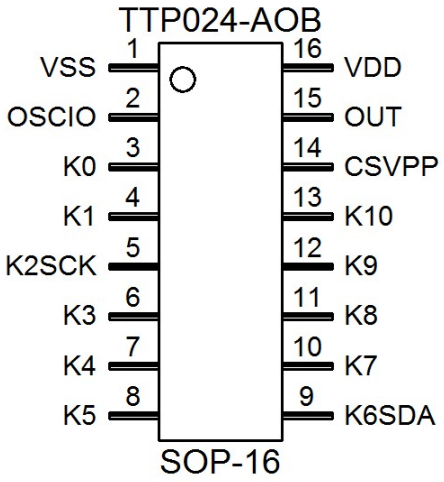
◆ Feature Description

- ◇ Tontek RISC 4-bit CPU core
- ◇ MTP structure
  - .2Kx16 ROM x1
  - .1Kx16 ROM x2
- ◇ Total 26 crucial instructions and two addressing mode
- ◇ Most instructions need 1 word and 1 machine cycle(2 system clocks) except read table instruction(RTB)
- ◇ advance CMOS process
- ◇ Working memory with 2K*16 program ROM and 64*4 SRAM
- ◇ 2-level stacks
- ◇ Low power consumption (VDD = 2.0V ~ 3.6V)
- ◇ PPM (Pulse Position Modulation) code method
- ◇ Build-in Internal Oscillator Frequency 455 KHz ± 2%
- ◇ 16 Custom Code (16 Bits) can be programmable
- ◇ 66 Key Data Code (8 Bits) can be programmable
- ◇ LED Display
- ◇ Used or No-used the Transistor [By Programming]
- ◇ Carry (Duty) or No-Carry [By Programming]
- ◇ Transmission Code Format [By Programming]
- ◇ SOP-16 Package (150 mil)

◆ Block Diagram



◆ Package Configuration



◆ Pin Description

PIN No. SOP-16	PIN Name	I/O	Description
1	VSS	P	Negative Power Supply
2	OSCIO	IO	Input Only for Test & Output can be used LED Display
3	K0/PA0	IO	Key-Scan/Custom code Setting
4	K1/PA1	IO	Key-Scan/Custom code Setting
5	K2/PA2/CLK	IO	Key-Scan/Custom code Setting & Programming CLK
6	K3/PA3	IO	Key-Scan/Custom code Setting
7	K4/PB0	IO	Key-Scan/Custom code Setting
8	K5/PB1	IO	Key-Scan/Custom code Setting
9	K6/PB2/SDA	IO	Key-Scan/Custom code Setting & Programming Data
10	K7/PB3	IO	Key-Scan/Custom code Setting
11	K8/PC0	IO	Key-Scan/Custom code Setting
12	K9/PC1	IO	Key-Scan/Custom code Setting
13	K10/PC2	IO	Key-Scan/Custom code Setting
14	CSVPP	I	Custom Code Setting by Application Programming High-Voltage Input
15	OUT	O	Serial Data Output
16	VDD	P	Positive power supply

◆ Absolute Maximum Ratings

Item Decription	Symbol	Ratings	Unit
Supply voltage	VDD	2.0 ~ 3.6	V
Operating temperature	Topr	−10 ~ 50	°C
Storage temperature	Tstg	−50 ~ 125	°C

◆ Electronic Characteristics(VDD=3V@25℃)

Item	Symbol	Condition	Min.	Typ.	Max.	Unit
Supply voltage	VDD		**2.0**	**3.0**	**3.6**	V
Stand-by current (Oscillator OFF)	ISB	VDD = 3.0 V	—	**1.0**	**2.5**	uA
Driving current (OUT)	IOH1	VDD = 3.0 V (VO = 1.5 V)	—	**- 7**	—	mA
Sinking current (OUT)	IOL1	VDD = 3.0 V (VO = 0.3 V)	—	**200**	—	mA
Driving current (K0D0 ~ K10CLK)	IOH2	VDD = 3.0 V (VO = 1.5 V)	—	**- 70**	—	uA
Sinking current (K0D0 ~ K10CLK)	IOL2	VDD = 3.0 V (VO = 1.5 V)	—	**3**	—	mA
Sinking current (OSCIO)	IOL3	VDD = 3.0 V (VO = 1.5 V)	**3**	—	—	mA
Internal Oscillator Frequency(455KHz)	Fosc		**445.9 (-2%)**	**455.0**	**464.1 (+2%)**	KHz

◆ Setting Custom Code(CC0 ~ CC15)

◇ CSVPP is connected to one of K0 ~ K10 that selects 1 of 11 Custom Code。Data(16 Bits) of “ Custom Code ” setting by Programming。

CSVPP =	K0	K1	K2	K3SCK	K4	K5	K6SDA	K7	K8	K9	K10
Custom	**CC0**	**CC1**	**CC2**	**CC3**	**CC4**	**CC5**	**CC6**	**CC7**	**CC8**	**CC9**	**CC10**
Custom0-H	CC0-H	CC1-H	CC2-H	CC3-H	CC4-H	CC5-H	CC6-H	CC7-H	CC8-H	CC9-H	CC10-H
Custom0-L	CC0-L	CC1-L	CC2-L	CC3-L	CC4-L	CC5-L	CC6-L	CC7-L	CC8-L	CC9-L	CC10-L
Custom1-H	CC1-H	**CC1-H**	CC3-H	**CC3-H**	CC5-H	**CC5-H**	CC7-H	**CC7-H**	CC9-H	**CC9-H**	CC11-H
Custom1-L	CC1-L	**CC1-L**	CC3-L	**CC3-L**	CC5-L	**CC5-L**	CC7-L	**CC7-L**	CC9-L	**CC9-L**	CC11-L

◇ CSVPP is connected to VDD or NC，Each Key selects 1 of 8 Custom Code respectively。Data(16 Bits) of “ Custom Code ” setting by Programming。

CSVPP =	**VDD 或 NC**							
Custom	**CC8**	**CC9**	**CC10**	**CC11**	**CC12**	**CC13**	**CC14**	**CC15**
Custom0-H	CC8-H	CC9-H	CC10-H	CC11-H	CC12-H	CC13-H	CC14-H	CC15-H
Custom0-L	CC8-L	CC9-L	CC10-L	CC11-L	CC12-L	CC13-L	CC14-L	CC15-L
Custom1-H	CC9-H	**CC9-H**	CC11-H	**CC11-H**	CC13-H	**CC13-H**	CC15-H	**CC15-H**
Custom1-L	CC9-L	**CC9-L**	CC11-L	**CC11-L**	CC13-L	**CC13-L**	CC15-L	**CC15-L**

◆ Setting Output Data of Custom Code

◇ Output of “ Custom Code ” ：(Select Output **1 ~ 32** Bits)

- High-Byte & Low-Byte be used
- High-Byte(Custom0-H & Custom1-H) ：Select Output **1 ~ 8** Bits or No-used
- Low-Byte(Custom0-L & Custom1-L) ：Select Output **1 ~ 8** Bits or No-used
- Output Data of Each Byte must be “ **Bit-0** ” → “ **Bit-7** ”

Custom	**CSVPP == K0 [CC0 == $F3A5 & CC1 == $3F5A]**							
	Bit-0	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
Custom0-H [$F3]	**1**	**1**	**0**	**0**	**1**	**1**	**1**	**1**
Custom0-L [$A5]	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**
Custom1-H [$3F]	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**
Custom1-L [$5A]	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**

Example：Custom0-H Select **6** Bits & Custom0-L Select **7** Bits，
Custom1-H Select **5** Bits & Custom1-L Select **6** Bits，
Custom Code：Select **24** Bits

Custom	**CSVPP == K0 [CC0 == $F3A5 & CC1 == $3F5A]**							
	Bit-0	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
Custom0-H [$33]	**1**	**1**	**0**	**0**	**1**	**1**		
Custom0-L [$25]	**1**	**0**	**1**	**0**	**0**	**1**	**0**	
Custom1-H [$1F]	**1**	**1**	**1**	**1**	**1**			
Custom1-L [$1A]	**0**	**1**	**0**	**1**	**1**	**0**		

◇ Output of “ Custom Inverse Code ” ：(Select Output **1 ~ 32** Bits)

- By inverting the “ Custom Code(16 Bits) ”
- High-Byte inverting & Low-Byte inverting be used
- High-Byte(Custom0B-H & Custom1B-H)：Select Output **1 ~ 8** Bits or No-used
- Low-Byte(Custom0B-L & Custom1B-L)：Select Output **1 ~ 8** Bits or No-used

  Example：Custom Code == $F3A5 then the Inverse Code == $0C5A
- Output Data of Each Byte must be “ **Bit-0** ” → “ **Bit-7** ”

Custom	CSVPP == K0 [CC0 == $F3A5 & CC1 == $3F5A]							
	Bit-0	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7
Custom0-H [$F3]	1	1	0	0	1	1	1	1
Custom0-L [$A5]	1	0	1	0	0	1	0	1
Custom1-H [$3F]	1	1	1	1	1	1	0	0
Custom1-L [$5A]	0	1	0	1	1	0	1	0

Custom-B	CSVPP == K0 [CC0-B == $0C5A & CC1-B == $C0A5]							
	Bit-0	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7
Custom0B-H [$0C]	0	0	1	1	0	0	0	0
Custom0B-L [$5A]	0	1	0	1	1	0	1	0
Custom1B-H [$C0]	0	0	0	0	0	0	1	1
Custom1B-L [$A5]	1	0	1	0	0	1	0	1

Example：Custom0B-H Select **6** Bits & Custom0B-L Select **7** Bits，
Custom1B-H Select **5** Bits & Custom1B-L Select **6** Bits，
Custom Code inverting：Select **24** Bits

Custom-B	CSVPP == K0 [CC0-B == $0C5A & CC1-B == $C0A5]							
	Bit-0	Bit-1	Bit-2	Bit-3	Bit-4	Bit-5	Bit-6	Bit-7
Custom0B-H [$0C]	0	0	1	1	0	0		
Custom0B-L [$5A]	0	1	0	1	1	0	1	
Custom1B-H [$00]	0	0	0	0	0			
Custom1B-L [$25]	1	0	1	0	0	1		

◆ Setting Data Code(66 Keys)

T-Matrix 66 Keys is compose of K0 ~ K10 and VSS(KL1 ~ KL66)，
Data(8 Bits) of “ Data Code ” setting by Programming。

	K10 (Pin-13)	K9 (Pin-12)	K8 (Pin-11)	K7 (Pin-10)	K6SDA (Pin-9)	K5 (Pin-8)	K4 (Pin-7)	K3SCK (Pin-6)	K2 (Pin-5)	K1 (Pin-4)	K0 (Pin-3)
K1 (Pin-4)											**KL1**
K2 (Pin-5)										**KL12**	**KL2**
K3SCK (Pin-6)									**KL22**	**KL13**	**KL3**
K4 (Pin-7)								**KL31**	**KL23**	**KL14**	**KL4**
K5 (Pin-8)							**KL39**	**KL32**	**KL24**	**KL15**	**KL5**
K6SDA (Pin-9)						**KL46**	**KL40**	**KL33**	**KL25**	**KL16**	**KL6**
K7 (Pin-10)					**KL52**	**KL47**	**KL41**	**KL34**	**KL26**	**KL17**	**KL7**
K8 (Pin-11)				**KL57**	**KL53**	**KL48**	**KL42**	**KL35**	**KL27**	**KL18**	**KL8**
K9 (Pin-12)			**KL61**	**KL58**	**KL54**	**KL49**	**KL43**	**KL36**	**KL28**	**KL19**	**KL9**
K10 (Pin-13)		**KL64**	**KL62**	**KL59**	**KL55**	**KL50**	**KL44**	**KL37**	**KL29**	**KL20**	**KL10**
VSS (Pin-1)	**KL66**	**KL65**	**KL63**	**KL60**	**KL56**	**KL51**	**KL45**	**KL38**	**KL30**	**KL21**	**KL11**

◆ Setting Output Data of Data Code

◇ Output of “ Data Code ” ：(Select Output **1 ~ 8** Bits)

- Data Code(Data)：Select Output **1 ~ 8** Bits or No-used
- Output Data of “ Data Code ” must be “ **Bit-0** ” → “ **Bit-7** ”

Data Code	Data(8 Bits) of the Data Code == $9A							
Data	**Bit-0**	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
	0	**1**	**0**	**1**	**1**	**0**	**0**	**1**

Example：Output Data of “ Data Code ” Select **7** Bits

Data Code	Output(7 Bits) == $1A							
Data	**Bit-0**	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	
	0	**1**	**0**	**1**	**1**	**0**	**0**	

◇ Output of “ Data Inverse Code ” ：(Select Output **1 ~ 8** Bits)

- By inverting the “ Data Code(8 Bits) ”
- Data Inverse Code(DataB)：Select Output **1 ~ 8** Bits or No-used

Example：Data Code == $9A，Data Inverse Code == $65

Data Code	Data(8 Bits) of the Inverse Code == $65							
DataB	**Bit-0**	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
	1	**0**	**1**	**0**	**0**	**1**	**1**	**0**

◆ Output Data of Data Code，2 Mode be Selecting

◇ Mode-1：(Data、DataB be used)

- Output of Data Code is **unconcerned with Custom Code**(16 Bits)
- Output Data(8 Bits) of Data Code be used by Programming
  Example：Data Code == $5A，Data Inverse Code == $A5
  Output Data of Both Select **7** Bits

Data Code	Data Code == $5A & Data Inverse Code == $A5 Output of Data(7 Bits) = $5A & Output of DataB(7 Bits) = $25							
	Bit-0	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
Data	0	1	0	1	1	0	1	0
Output	**0**	**1**	**0**	**1**	**1**	**0**	**1**	
DataB	1	0	1	0	0	1	0	1
Output	**1**	**0**	**1**	**0**	**0**	**1**	**0**	

◇ Mode-2：(Data-C0、DataB-C0、Data-C1、DataB-C1 be used)

- Output Data of Data Code **depends on Custom Code**(16 Bits)，
  it is **unconcerned with Output Fotmat of Custom Code**
- Custom Code：When High-Byte & Low-Byte are the same Bit，
  that Bits are the same Data then the Data Code Bit is changeless(X)；
  and Bits are the different Data then the Data Code Bit is inverted
- Example：
  Custom Code**[CC0 == $25E6 & CC1 == $526E]**

Custom	CSVPP == K0 [CC0 == $25E6 & CC1 == $526E]							
	Bit-0	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
Custom0-H [$25]	***1***	***0***	1	0	0	1	***0***	***0***
Custom0-L [$E6]	***0***	***1***	1	0	0	1	***1***	***1***
Change	**Invert**	**Invert**	**X**	**X**	**X**	**X**	**Invert**	**Invert**
Custom1-H [$52]	0	1	***0***	***0***	***1***	***0***	1	0
Custom1-L [$6E]	0	1	***1***	***1***	***0***	***1***	1	0
Change	**X**	**X**	**Invert**	**Invert**	**Invert**	**Invert**	**X**	**X**

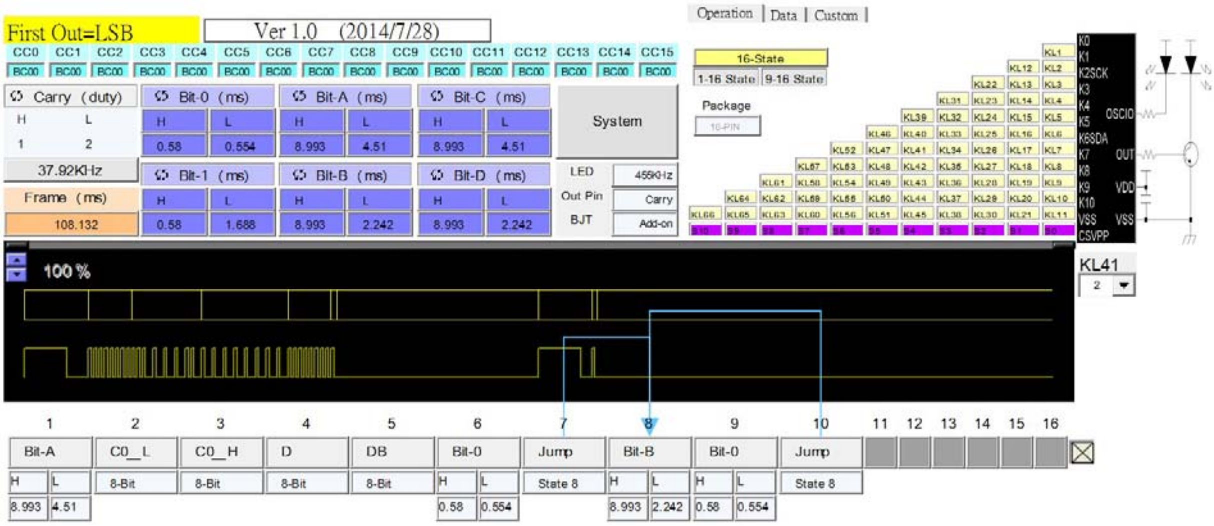
Data	CSVPP == K0 [CC0 == $25E6 & CC1 == $526E & Data==$5A]							
	Bit-0	**Bit-1**	**Bit-2**	**Bit-3**	**Bit-4**	**Bit-5**	**Bit-6**	**Bit-7**
Data [$5A]	0	1	0	1	1	0	1	0
Change [CC0]	**Invert**	**Invert**	**X**	**X**	**X**	**X**	**Invert**	**Invert**
Data-C0[$99]	***1***	***0***	0	1	1	0	***0***	***1***
Data [$5A]	0	1	0	1	1	0	1	0
Change [CC1]	**X**	**X**	**Invert**	**Invert**	**Invert**	**Invert**	**X**	**X**
Data-C1[$66]	0	1	***1***	***0***	***0***	***1***	1	0

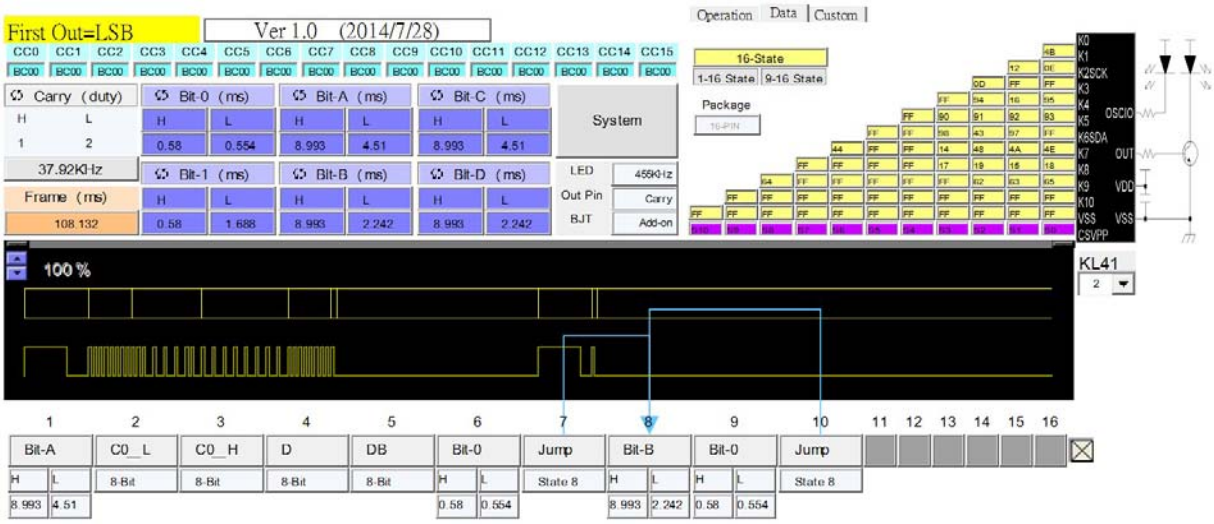
- DataB(8 Bits) must be inverted Data(8 Bits)
- DataB-C0(8 Bits) must be inverted Data-C0(8 Bits)
- DataB-C1(8 Bits) must be inverted Data-C1(8 Bits)

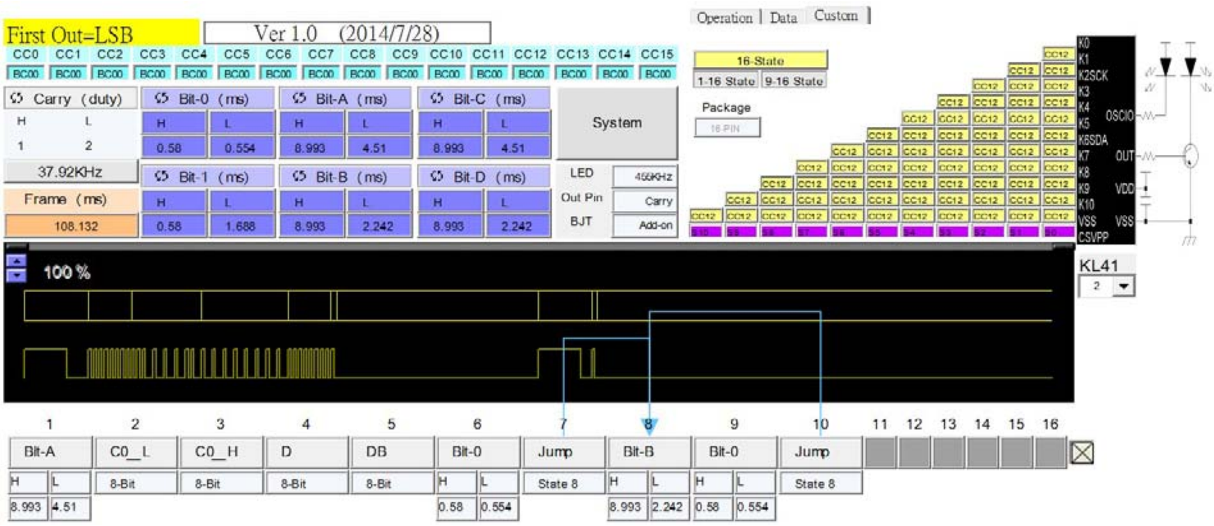
◆ Transmission Code Format

Tontek supplies customer with Software (Excel) and Hardware (Writer)

Excel Diagram(Software)：







◆ Setting the Transmission Code Format

◇ Setting the Timing of Bit-A、Bit-B、Bit-C、Bit-D、Bit-0 and Bit-1

- Bit-A、Bit-B、Bit-C、Bit-D、Bit-0 and Bit-1 are the same Timing & Form
- Output Timing by TA、TB are the same Timing & Form
- Timing of TA、TB：Min. time is 2 unit，Max. time is 511 unit
  unit time(26.4 us) is 12 CLK[455 KHz]
- Form of TA、TB：Select H-level(H) or L-level(L) respectively
- Example：6122 Form(TA + TB)

  Bit-A(Lead) ：H(=9.00 ms) + L(=4.50 ms)
  Bit-B(Repeat) ：H(=9.00 ms) + L(=2.25 ms)
  Bit-0 ：H(=0.56 ms) + L(=0.56 ms)
  Bit-1 ：H(=0.56 ms) + L(=1.68 ms)

◇ Setting One Frame of the Transmission Code

- Max. time of one Frame is 269.8 ms
- Example：6122 Form is 108 ms

◇ Setting Form of the Transmission Code by State Machine

- State Machine has **16** State to program
- Each State has **25** Form to program：
  “ Bit-A ” 、 “ Bit-B ” 、 “ Bit-C ” 、 “ Bit-D ” 、
  “ Bit-0 ” 、 “ Bit-1 ” 、 “ Custom0-H ” 、 “ Custom0-L ” 、
  “ Custom0B-H ” 、 “ Custom0B-L ” 、 “ Custom1-H ” 、
  “ Custom1-L ” 、 “ Custom1B-H ” 、 “ Custom1B-L ” 、
  “ Data ” 、 “ DataB ” 、 “ Data-C0 ” 、 “ DataB-C0 ” 、
  “ Data-C1 ” 、 “ DataB-C1 ” 、 “ Jump ” 、 “ Goto ” 、
  “ TG-AB ” 、 “ TG-CD ” 、 “ TG-01 ”
- “ Jump ” ：Hold-Key Transmits this Frame then transmits next Frame，
  Release-Key Transmits this Frame then ends transmission
  “ Goto ” ：Hold-Key or Release-Key Transmits this Frame
  then transmits next Frame
- Toggle(TG-AB & TG-CD & TG-01)：

  TG-AB ：Bit-A ⟷ Bit-B
  TG-CD ：Bit-C ⟷ Bit-D
  TG-01 ：Bit-0 ⟷ Bit-1

◇ Used or No-used the Transistor By Programming

◇ Output Form Carry or No-Carry By Programming

◇ Output Form Carry & Duty By Programming

- Carry H-level Width：1 ~ 127CK(CK==3.64MHz)[0.2747 ~ 34.8901us]
- Carry L-level Width：1 ~ 127CK(CK==3.64MHz)[0.2747 ~ 34.8901us]
- Carry Frequency：1.82MHz ~ 14.33KHz

◇ LED Display

- Output Display：System Frequency == 455KHz
- Output Display：Output Form By No-Carry

◆ Select the Transmission Code Format

◇ Single Format：(State Machine has **16** States to program)

● Example：6122 Form[Jump be used]

State	1	2	3	4	5	6	7	8
	Bit-A	**Custom0-H**	**Custom0-L**	Data	DataB	Bit-0	**Jump**	
Form	**Lead**	8 BIT	8 BIT	8 BIT	8 BIT	**End Bit**	**State 9**	
State	9	10	11	12	13	14	15	16
	Bit-B	Bit-0	Jump					
Form	**Repeat**	**End Bit**	**State 9**					

● Example：0773 Form-A[Jump & Goto be used]

State	1	2	3	4	5	6	7	8
	Custom0-L	Data	Bit-1	Bit-0	Bit-0	**Goto**		
Form	5 BIT	8 BIT			**End Bit**	**State 9**		
State	9	10	11	12	13	14	15	16
	Custom0-L	DataB	Bit-0	Bit-1	Bit-0	**Jump**		
Form	5 BIT	8 BIT			**End Bit**	**State 1**		

● Example：0773 Form-B[Jump & Goto be used]

State	1	2	3	4	5	6	7	8
	Custom0-L	Data	***Custom0-H***	Bit-0	**Goto**			
Form	5 BIT	8 BIT	2 BIT	**End Bit**	**State 9**			
State	9	10	11	12	13	14	15	16
	Custom0-L	DataB	***Custom0B-H***	Bit-0	**Jump**			
Form	5 BIT	8 BIT	2 BIT	**End Bit**	**State 1**			

● State Machine has **16** States to program by the need of customer
Example：0773 Form-A or 0773 Form-B

◇ Twin Formats：(State Machine has **16** States or **8** States to program)

● Select Twin Formats by programming

● The Bit-A、Bit-B、Bit-C、Bit-D、Bit-0、Bit-1、Frame are the same Form；CSVPP is connected to one of K0 ~ K10 that selects 1 of 11 Custom Code

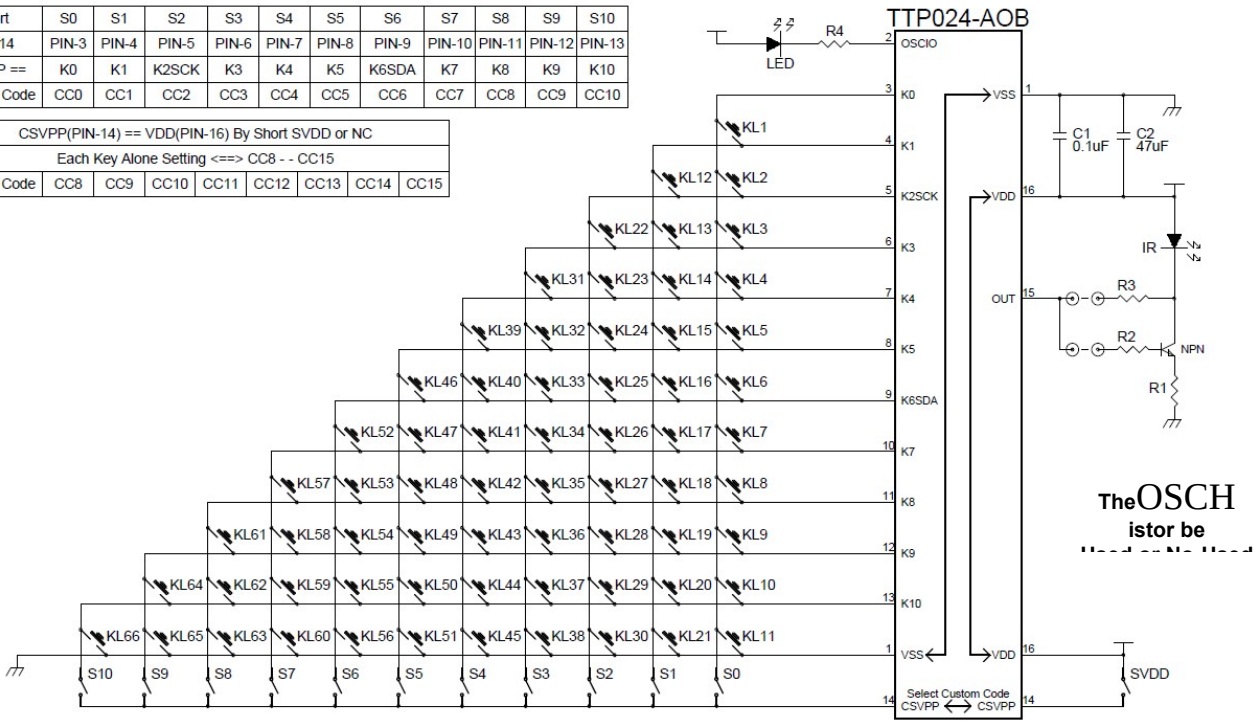
● CSVPP is connected to VDD or NC，Each Key can select 1 of 8 Custom Code respectively and setting the state

CSVPP ==	**VDD 或 NC**							
Custom	**CC8**	**CC9**	**CC10**	**CC11**	**CC12**	**CC13**	**CC14**	**CC15**
Select	**State 1 ~ 16**				**State 9 ~ 16**			

◆ Application Circuit [SOP-16 for 66-Key]

Short	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
PIN-14	PIN-3	PIN-4	PIN-5	PIN-6	PIN-7	PIN-8	PIN-9	PIN-10	PIN-11	PIN-12	PIN-13
CSVPP ==	K0	K1	K2SCK	K3	K4	K5	K6SDA	K7	K8	K9	K10
Custom Code	CC0	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10

CSVPP(PIN-14) == VDD(PIN-16) By Short SVDD or NC								
Each Key Alone Setting <==> CC8 - - CC15								
Custom Code	CC8	CC9	CC10	CC11	CC12	CC13	CC14	CC15



Note：1. Application Circuit Reference Only，Refer to real operating

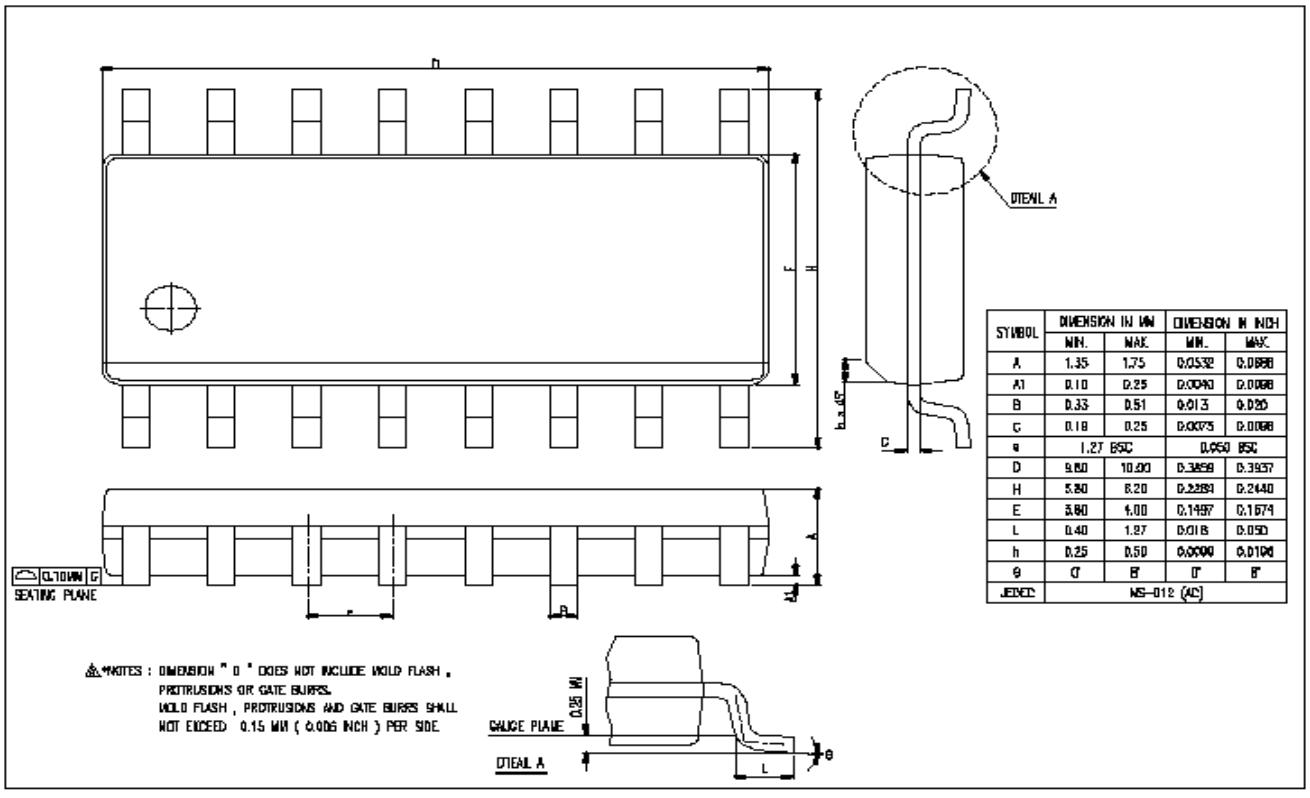
2. The Capacitor (C1 and C2) must be near to IC's Power pin [VDD and VSS]

3. The IC's Power must be near to Power-Supply by Power-Line Routing [PCB]

4. The value of Resistance(R1~R4) be adjusted by oneself，Refer to real operating

[By The mission distance and The LED brightness]

◆ SOP-16(150 mil) Outline



SYMBOL	DIMENSION IN MM		DIMENSION IN INCH	
	MIN.	MAX.	MIN.	MAX.
A	1.35	1.75	0.0532	0.0688
A1	0.10	0.25	0.0040	0.0098
B	0.33	0.51	0.013	0.020
C	0.18	0.25	0.0075	0.0098
e	1.27 BSC		0.050 BSC	
D	9.80	10.00	0.3858	0.3937
H	5.80	6.20	0.2283	0.2440
E	3.80	4.00	0.1497	0.1574
L	0.40	1.27	0.016	0.050
h	0.25	0.50	0.0099	0.0196
θ	0°	8°	0°	8°
JEDEC	MS-012 (AC)			

Ordering Information

TTP024		
Package Type	**Chip Type**	**Wafer Type**
TTP024-AOB(SOP-16)	No support	TDP024

◆ REVISE HISTORY

1. 2014/07/28
   -Original Version：Ver 1.0
2. 2014/10/10
   -Modify Page.3 Pin Description