4-Channel Programmable RGBW-LED Driver

Features

- 4-channel programmable LED driver individually
  - 25.5mA Imax
  - Individual 8-bit current resolution
  - Individual 8-bit PWM
- 3 program execution engines With Flexible Instruction Set
- Autonomous Operation With Program Execution Engines
- SRAM memory for lighting pattern programs
- Current accuracy: ±4%@17.5mA
- Current match accuracy: 2%@17.5mA
- Auto power save mode
- 1MHz I²C Interface
- Support 1.2V/1.8V logic level
- Power supply: 2.7V~5.5V
- AW21104FOR: FOWLP 1.62mmX1.22mm-12B

Applications

Fun Lights
Indicator Lights
RGB Backlighting

General Description

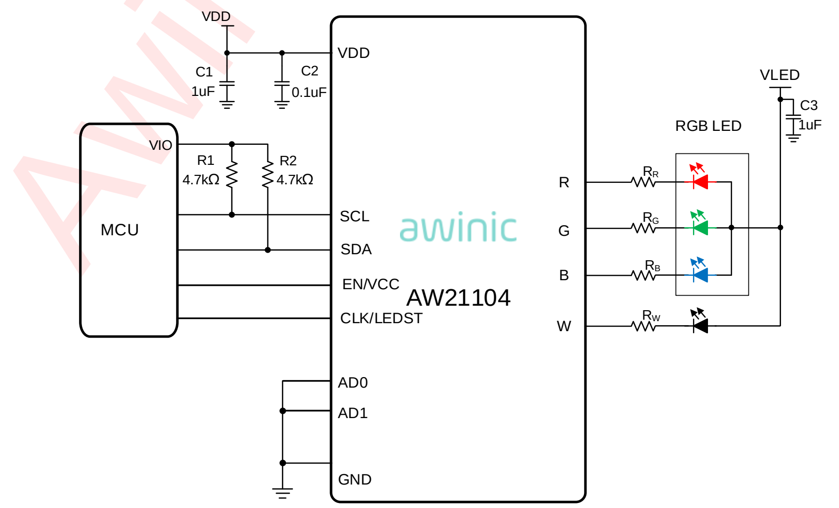
The AW21104FOR is a 4-channel LED driver designed to produce variety of lighting effects. The device has a program memory for creating variety of lighting sequences. When the program memory has been loaded, the AW21104FOR can operate independently without processor control.

The AW21104FOR is able to automatically enter power save mode, when LED outputs are not active and thus lowering current consumption.

Four independent LED channels have accurate programmable current sinks, from 0mA to 25.5mA with 100μA steps and flexible PWM control. Each channel can be configured into each of the three program execution engines. Program execution engines have program memory for creating desired lighting sequences with PWM control.

The AW21104FOR has four pin-selectable I²C addresses. This allows connecting up to four parallel devices in one I²C bus. The device requires only one small, low-cost ceramic capacitor.

Typical Application Circuit

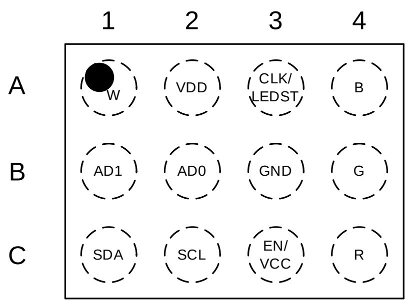


Note: The resistor R_{LED} is only thermal reduction, and it is determined by V_{LED}, VF of LED, $V_{DROPOUT}$ of LEDx and I_{LED}.
$R_{LED}=(V_{LED}-VF-V_{DROPOUT})/I_{LED}$.

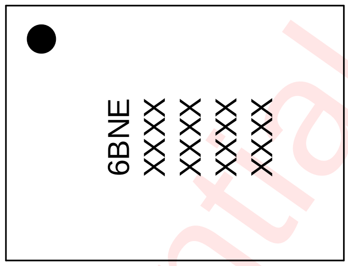
Figure 1 AW21104FOR Simple Application Circuit

Pin Configuration And Top Mark

AW21104FOR
(Top view)



AW21104FOR Marking
(Top view)



6BNE - AW21104FOR
XXXX/XXXX/XXXX/XXXX - Production Tracing Code

Figure 2 Pin Configuration and Marking

Pin Definition

NO. AW21104FOR	Name	Description
A1	W	LED driver current sink terminal
A2	VDD	Power supply
A3	CLK/LEDST	External 32kHz clock input or Internal 32kHz clock output or LED Status Indication output
A4	B	LED driver current sink terminal
B1	AD1	I^{2}C address selection pin 1
B2	AD0	I^{2}C address selection pin 0
B3	GND	Ground
B4	G	LED driver current sink terminal
C1	SDA	I^{2}C serial interface data input/output
C2	SCL	I^{2}C serial interface clock
C3	EN/VCC	Enable/Logic power supply
C4	R	LED driver current sink terminal
	Thermal pad	Only for heat dissipation

Functional Block Diagram

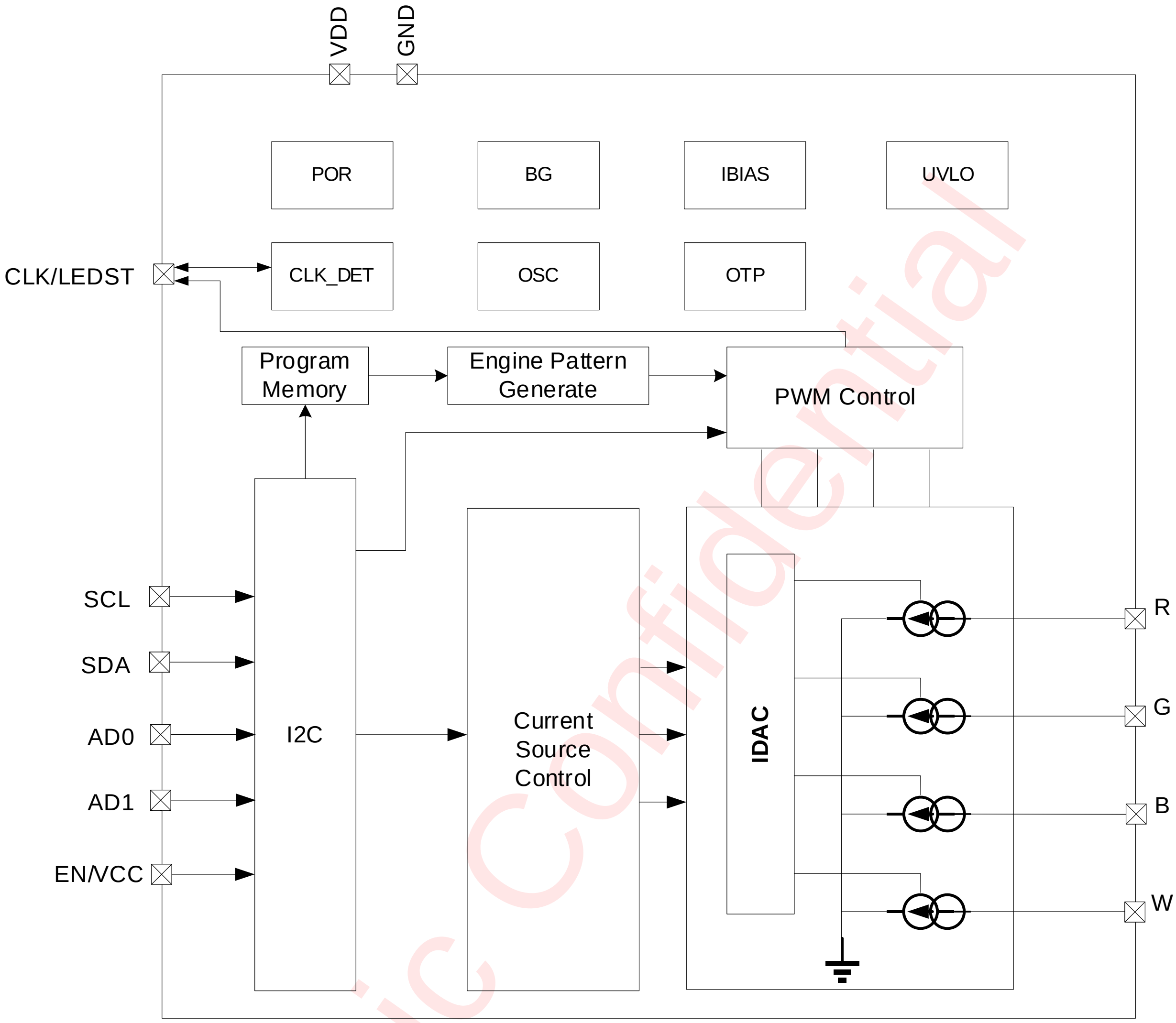


Figure 3 Functional Block Diagram

Ordering Information

Part Number	Temperature	Package	Marking	Moisture Sensitivity Level	Environmental Information	Delivery Form
AW21104FOR	-40°C~105°C	FOWLP 1.62mmX1.22mm-12B	6BNE	MSL1	RoHS+HF	3000 units/ Tape and Reel

Absolute Maximum Ratings(NOTE1)

PARAMETERS		RANGE
Supply voltage range VDD		-0.3V to 6V
Input voltage range	SCL, SDA, EN/VCC, CLK/LEDST, AD0, AD1	-0.3V to VDD
Output voltage range	W, B, G, R	-0.3V to VDD
Junction-to-ambient thermal resistance θ_{JA}		36.4°C/W
Operating free-air temperature range		-40°C to 105°C
Maximum operating junction temperature T_{JMAX}		125°C
Storage temperature T_{STG}		-65°C to 150°C
Lead temperature (soldering 10 seconds)		260°C
ESD(Including CDM HBM)(NOTE2)		
HBM		±4kV
CDM		±1.5kV
Latch-Up		
Test condition: JESD78F		+IT: 300mA -IT: -300mA

NOTE1: Conditions out of those ranges listed in "absolute maximum ratings" may cause permanent damages to the device. In spite of the limits above, functional operation conditions of the device should within the ranges listed in "recommended operating conditions". Exposure to absolute-maximum-rated conditions for prolonged periods may affect device reliability.

NOTE2: The HBM test method: ESDA/JEDEC JS-001-2023, the CDM test method: ESDA/JEDEC JS-002-2022.

Recommended Operating Conditions

Symbol	Parameters	Min.	Typ.	Max.	Unit
VDD	Supply voltage	2.7		5.5	V
$V_{EN/VCC}$	input voltage range	1.2		VDD	V
T_J	Junction temperature	-40		125	°C
T_A	Operating free-air temperature	-40		105	°C

Electrical Characteristics

Unless otherwise specified: limits for typical values are for T_A=25°C and minimum and maximum limits apply over the operating ambient temperature range (−40°C<T_A<105°C); VDD=3.6V, $V_{EN/VCC}$=1.8V.

Parameter		Test Condition	Min.	Typ.	Max.	Unit
Power supply voltage and current						
VDD	Power supply voltage		2.7	3.6	5.5	V
$V_{EN/VCC}$			1.2		VDD	V
I_{STB1}	Standby supply current	EN=0(pin), CHIP_EN=0(bit), external 32kHz clock running or not running	0.02	0.24	2	μA
I_{STB2}		EN=1(pin), CHIP_EN=0(bit), external 32kHz clock not running	0.3	1	2.5	μA
I_{STB3}		EN=1(pin), CHIP_EN=0(bit), external 32kHz clock running	0.3	1.4	3	μA
I_{Q1}	Normal mode supply current	EN=1(pin), CHIPEN=1(bit), LED disable(All LED control register is 0)	0.1	0.15	0.2	mA
I_{Q2}		EN=1(pin), CHIPEN=1(bit), LED output enable, PWM = 0 Note1	0.2	0.3	0.4	mA
		EN=1(pin), CHIPEN=1(bit), LED output enable, PWM≠0	0.3	0.8	1.2	mA
I_{Q3}	Power-save mode supply current	EN=1(pin), CHIPEN=1(bit), power save triggered and external 32kHz clock running	0.2	2	10	μA
I_{Q4}		EN=1(pin), CHIPEN=1(bit), power save triggered and internal oscillator running	0.1	0.15	0.2	mA
$I_{LEAKAGE}$	R, G, B, W pin leakage current	T_A = 25°C	0	0.1	1	μA
I_{MAX}	R, G, B, W output maximum current	All LED current control register is 0XFF	24.2	25.5	26.8	mA
I_{OUT}	R, G, B, W output current accuracy	Output current set to 17.5mA (R/G/B/W_CURRENT=0xAF), VDD=3.6V, T_A=25°C	-4		+4	%
		Output current set to 17.5mA (R/G/B/W_CURRENT=0xAF) VDD=3.6V	-5		+5	%
I_{MATCH}	R, G, B, W current matching ((MAX-AVG) /AVG OR (AVG-MIN)/AVG)	Output current set to 17.5mA (R/G/B/W_CURRENT=0xAF) , VDD=3.6V	0	1	2	%

Parameter		Test Condition	Min.	Typ.	Max.	Unit
$V_{DROPOUT}$	R, G, B, W pin voltage when the output current has dropped 10% from the set current	Output current set to 17.5mA (R/G/B/W_CURRENT = 0xAF), T_A=25°C	2	60	100	mV
F_{PWM}	LED PWM switching frequency		128		4.09k	Hz
F_{OSC}	Internal OSC clock frequency	T_A = 25°C	-3%	1.048	3%	MHz
		T_A = -40~105°C	-4%	1.048	4%	MHz
POR	Power on reset threshold voltage	VDD<POR ,chip is reset	1.8	2.1	2.4	V
POR_{HYS}	Power on reset hysteresis voltage		0.25	0.3	1.05	V
OTP Note1	Over-temperature protection threshold		130	150	170	°C
OTP_{HYS} Note1	Over-temperature protection hysteresis		15	20	25	°C
LOGIC INTERFACE CHARACTERISTICS(EN/VCC)						
V_{IH}	Input high level		1.08		VDD	V
V_{IL}	Input low level		0		0.4	V
I_I	Input current	$V_{EN/VCC}$=1.2V~VDD	-1		1	μA
t_{DELAY} Note1,2	Input delay		1	2	10	μs
LOGIC INTERFACE CHARACTERISTICS(SCL,SDA,CLK/LEDST,AD0,AD1)						
$V_{OL,SDA}$ Note1	Output low level of SDA pin	SDA pin Iout=3mA (pullup current)	0	0.3	0.5	V
$I_{L,SDA}$	Output leakage current of SDA pin		-1		1	μA
V_{IH}	Input high level		0.8× $V_{EN/VCC}$		$V_{EN/VCC}$	V
V_{IL}	Input low level		0		0.2× $V_{EN/VCC}$	V
I_I	Input current	$V_{EN/VCC}$=1.2V~VDD	-1		1	μA
F_{CLK_32K}			31	32	33	kHz

Note1: Minimum and/or maximum limit is guaranteed by design and by statistical analysis of device characterization data. The specification is not guaranteed by production testing.

Note2: The I^2C host should allow at least 1ms before sending data to the AW21104FOR after the rising edge of the enable line.

I²C Interface Timing Requirements

Parameter		Fast Mode		Fast Mode Plus		Unit
		Min.	Max.	Min.	Max.	
F_{SCL}	Interface clock frequency	-	400	-	1000	kHz
$T_{HD:STA}$	(Repeat-start) START condition hold time	0.6	-	0.26	-	μs
T_{LOW}	Low level width of SCL	1.3	-	0.5	-	μs
T_{HIGH}	High level width of SCL	0.6	-	0.26	-	μs
$T_{SU:STA}$	(Repeat-start) START condition setup time	0.6	-	0.26	-	μs
$T_{HD:DAT}$	Data hold time	0	-	0	-	μs
$T_{SU:DAT}$	Data setup time	0.1	-	0.05	-	μs
T_R	Rising time of SDA and SCL	-	0.3	-	0.12	μs
T_F	Falling time of SDA and SCL	-	0.3	-	0.12	μs
$T_{SU:STO}$	STOP condition setup time	0.6	-	0.26	-	μs
T_{BUF}	Time between start and stop condition	1.3	-	0.5	-	μs

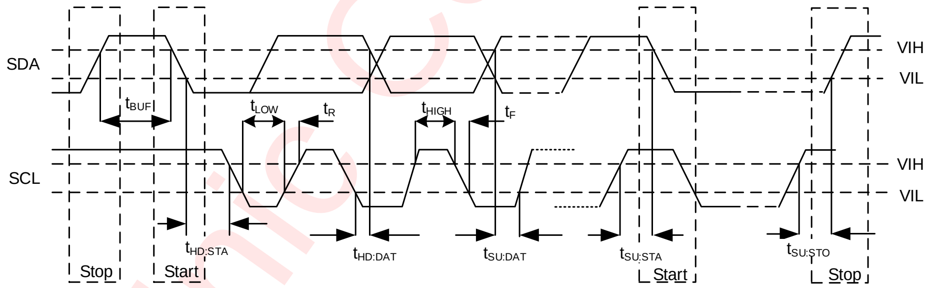


Figure 4 I²C Interface Timing

Detailed Functional Description

The AW21104FOR is a RGBW LED driver for indicator LED and keypad lighting. The device has an internal program memory for creating a variety of lighting sequences. When the program memory has been loaded, the AW21104FOR can operate independently without processor control.

POWER ON RESET

Upon initial power-up, the AW21104FOR is reset by internal power-on-reset, and all registers are reset to default value, and the chip is shut down.

Once VDD rises above 2.1V(typ.), POR will inactivate and the chip will continue to the standby mode. EN/VCC power on no later than VDD. CHIP_EN control bit is low after POR by default.

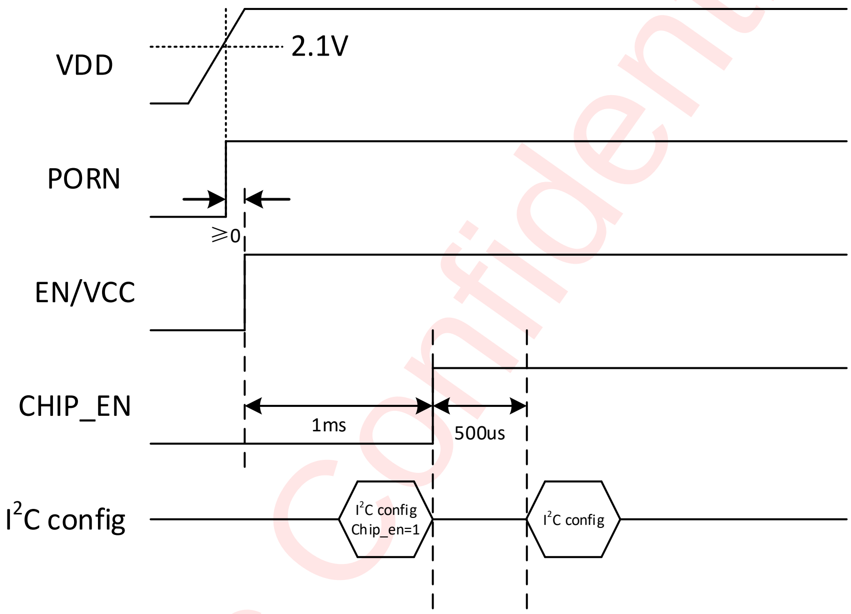


Figure 5 power on timing

DEVICE FUNCTIONAL MODES

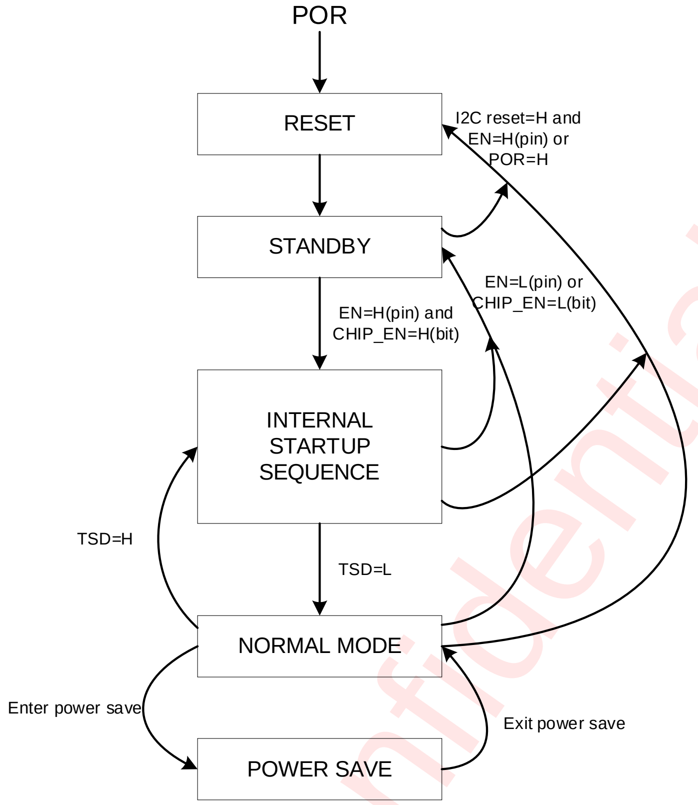


Figure 6 perating mode transition

RESET MODE

Reset is done always if 0xFFh is written to Reset Register (address 16h) or internal Power On Reset is activated. In the reset mode all the internal registers are reset to the default values. Power On Reset (POR) will activate when supply voltage is connected or when the supply voltage VDD falls below 1.9V (typ.). Once VDD rises above 2.1V (typ.), POR will inactivate and the chip will continue to the standby mode. CHIP_EN control bit is low after POR by default.

STANDBY MODE

The standby mode is entered if the register bit CHIP_EN or EN pin is low and Reset is not active. This is the low power consumption mode, when all circuit functions are disabled. Registers can be written in this mode if EN pin is high. Control bits are effective after start up.

STARTUP MODE

When CHIP_EN bit is written high and EN pin is high, the internal startup sequence powers up all the needed internal blocks (VREF, Bias, Oscillator etc.). Startup delay after setting EN pin high is 1ms (typ.). Startup delay after setting CHIP_EN bit to '1' is 500μs (typ.). If the device temperature rises too high, the Thermal Shutdown (TSD) disables the device operation and the device state is in startup mode, until no thermal shutdown event is present.

NORMAL MODE

During normal mode the user controls the device using the Control Registers. If EN pin is set low, the CHIP_EN bit is reset to 0.

POWER-SAVE MODE

In power save mode analog blocks are disabled to minimize power consumption. See chapter Power Save Mode for further information.

LED DRIVERS OPERATIONAL DESCRIPTION

The AW21104FOR has 4 LED drivers, and the current of each driver can be controlled through an 8-bit current source or 8-bit PWM separately. Current is controlled from I^2C registers. PWM can be controlled with program execution engines or direct I^2C register writes.

LED DRIVER CURRENT CONTROL

LED driver output current can be programmed with I^2C register from 0mA up to 25.5mA. Current setting resolution is 100μA (8-bit control).

Table 1 B_CURRENT Register (11h), G_CURRENT Register (12h), R_CURRENT Register (13h), W_CURRENT Register (14h)

Name	Bits	Description			
		Current setting			
		bin	hex	dec	mA
		0000 0000	00	0	0.0
		0000 0001	01	1	0.1
		0000 0010	02	2	0.2
		0000 0011	03	3	0.3
		0000 0100	04	4	0.4
		0000 0101	05	5	0.5
CURRENT	7:0	0000 0110	06	6	0.6
		…	…	…	…
		10101111	AF	175	17.5(def)
		…	…	…	…
		1111 1011	FB	251	25.1
		1111 1100	FC	252	252
		1111 1101	FS	253	25.3
		1111 1110	FE	254	25.4
		1111 1111	FF	255	25.5

PWM FREQUENCY SELECTION

The user can select the desired PWM frequency by configuring the PWM_FRQ of CONFIG1 (06h), configurable PWM frequencies are as follows:

Table 2 PWM frequencies Register (06h)

PWM_FRQ[2:0]	0	1	2	3	4	5
PWM frequency	4.09KHz	2.05KHz	1.02KHz	512Hz	256Hz	128Hz

PWM LOG MODULATION

In order to make the brightness changes smoother for humans, this device adds log modulation function. Users can enable this function by configuring the LOG_EN of CONFIG1 (06h).

PWM defaults to linear variation, and the log modulation of an 8-bit PWM can only choose normal fitting(log1), The relationship between PWM value and duty cycle is shown in the following figure:

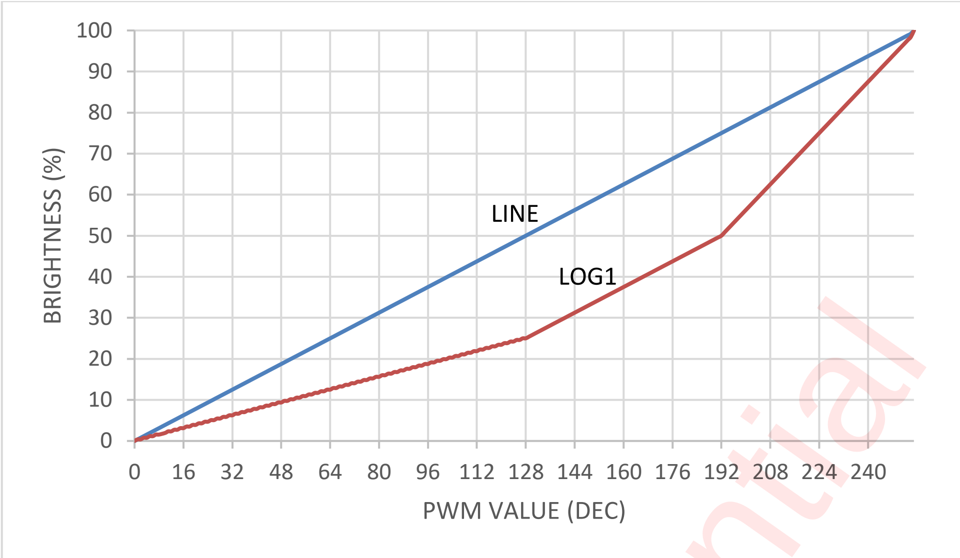


Figure 7 Logarithmic and Linear PWM Adjustment Curves

PWM CONTROL MODE

The PWM of AW21104FOR can be controlled in the following two ways:

1. I²C register (Four LED channels correspond to four sets of PWM control registers, 09h for B, 0Ah for G, 0Bh for R, 0Ch for W);
2. Three internal program engines (Engine1, Engine2, Engine3).

The user can select the control mode of the 4-channel LED driver by configuring the LED_MAP (04h) register, as follows:

Table 3 LED PWM Output Selection Bits

B_ENG_SEL bits[1:0] G_ENG_SEL bits[3:2] R_ENG_SEL bits[5:4] W_ENG_SEL bits[7:6]	Description
00	Output is controlled via I²C registers
01	ENG1_MODE and ENG1_EXEC register control LED output PWM instead of I²C register
10	ENG2_MODE and ENG2_EXEC register control LED output PWM instead of I²C register
11	ENG3_MODE and ENG3_EXEC register control LED output PWM instead of I²C register

If the LED driver output is controlled by an internal engine, the internal engine will control the PWM output by executing a program. However, when the engine output is set to direct mode by configuring the OP-MODE (03h) , the corresponding engine output will be directly controlled by the I²C register.

The PWM control logic corresponding to four LED channels in different modes is as follows:

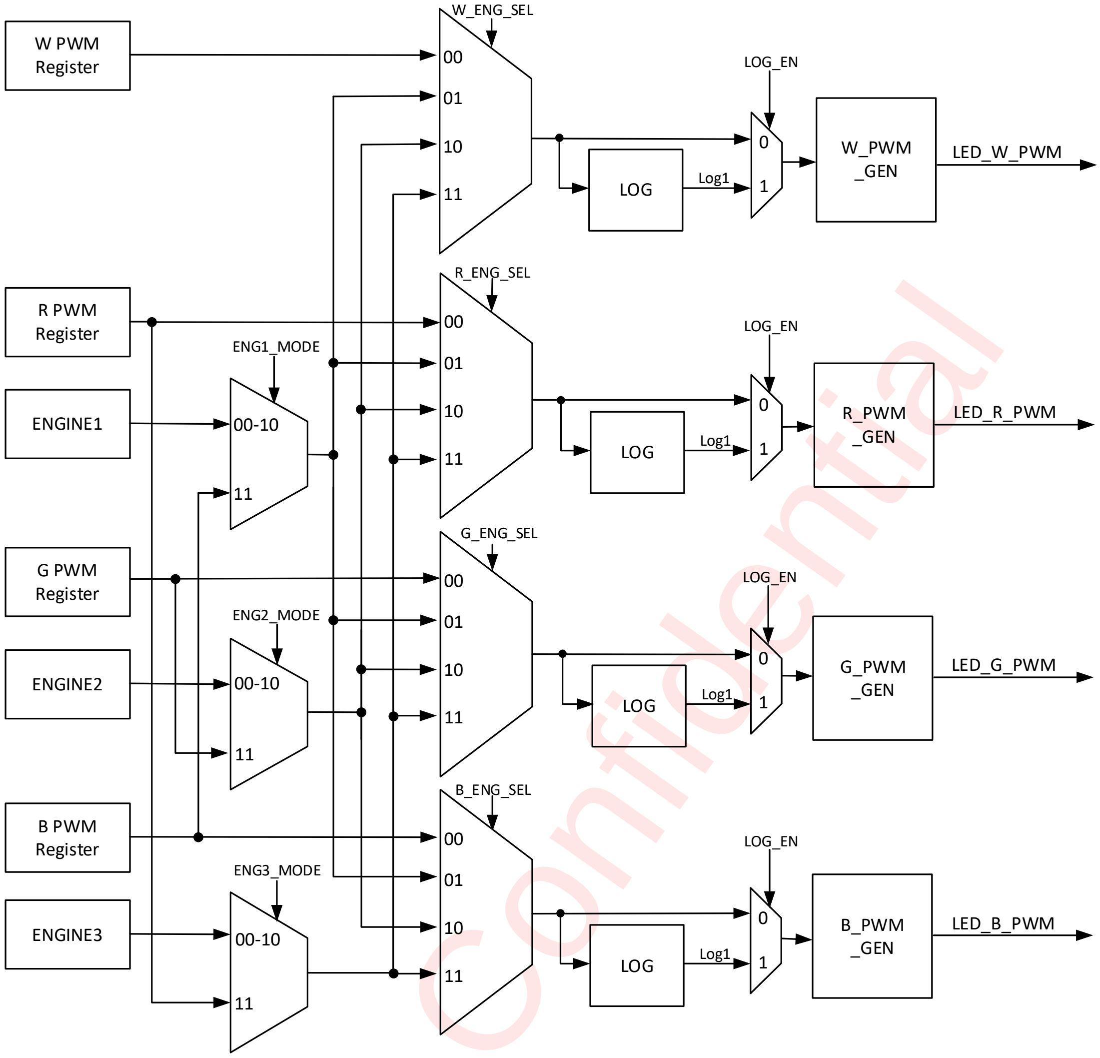


Figure 8 Logarithmic and Linear PWM Adjustment Curves

DIRECT I²C REGISTER PWM CONTROL

PWM can directly control LED drivers through I²C read/write registers.

The specific steps are as follows:

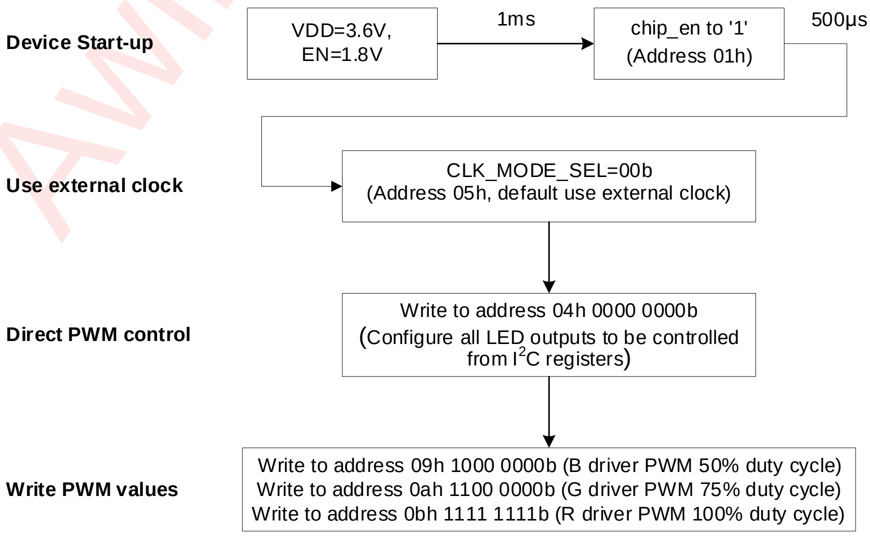


Figure 9 Direct I²C Register PWM Control Flow

LED STATUS INDICATION

Status indicator signal is set to 1 when LED is running. There are four status register to indicate LEDs in register address 17h. The result about OR operation of four LED status can be sent out by CLK/LEDST pin when CLK32K_DOUT_SEL bit in register address 05h is 0 and CLK32K_OBE bit in register address 05h is 1. Each LED has a enable bit to control whether the status indicator signal is being sent or not in the 08h register. For example, The status indicator of B LED can not be sent out by CLK/LEDST pin when the LED_B_ST_EN bit in register address 08h is 0.

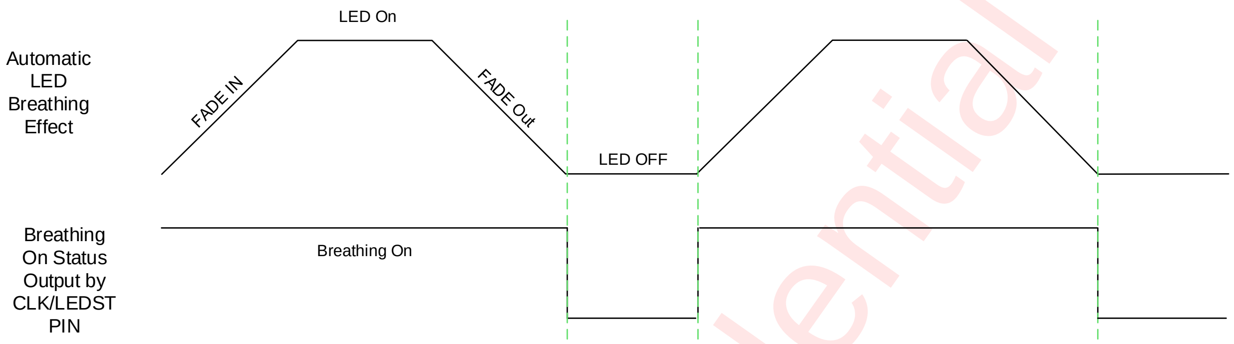


Figure 10 LED Status Indication Diagram

PROGRAMMING

SRAM MEMORY

In the AW21104FOR there is a SRAM memory reserved for storing the LED lighting programs. Each engine has its own section of the memory so that engine 1 has registers 20h to 3Fh, engine 2 has registers 40h to 5Fh, and engine 3 has registers 60h to 7Fh. For each engine 16 engine commands (16-bit) can be stored. Each 16-bit command takes up two I²C registers.

Table 4 SRAM Memory Registers

ADDR	Name	R/W	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Default
0x20	PROG MEM ENG1	RW	CMD1_ENG1[15:8]								0x00
0x21	PROG MEM ENG1	RW	CMD1_ENG1[7:0]								0x00
...											
0x3E	PROG MEM ENG1	RW	CMD16_ENG1[15:8]								0x00
0x3F	PROG MEM ENG1	RW	CMD16_ENG1[7:0]								0x00
...											
0x40	PROG MEM ENG2	RW	CMD1_ENG2[15:8]								0x00
0x41	PROG MEM ENG2	RW	CMD1_ENG2[7:0]								0x00
...											
0x5E	PROG MEM ENG2	RW	CMD16_ENG2[15:8]								0x00
0x5F	PROG MEM ENG2	RW	CMD16_ENG2[7:0]								0x00

...				
0x60	PROG MEM ENG3	RW	CMD1_ENG3[15:8]	0x00
0x61	PROG MEM ENG3	RW	CMD1_ENG3[7:0]	0x00
...				
0x7E	PROG MEM ENG3	RW	CMD16_ENG3[15:8]	0x00
0x7F	PROG MEM ENG3	RW	CMD16_ENG3[7:0]	0x00

PROGRAM EXECUTION ENGINES

Use of program execution engines is the other LED output PWM control method available in the AW21104. The device has 3 program execution engines. These engines create PWM controlled lighting patterns to the mapped LED outputs according to program codes developed by the user. Program coding is done using programming commands. Programs are loaded into SRAM memory and engine control bits are used to run these programs autonomously. LED outputs can be mapped into these 3 engines with register 04h bit settings. The engines have different operation modes, program execution states, and program counters. Each engine has its own section of the SRAM memory.

PROGRAM EXECUTION ENGINE MODES

Engine program execution is controlled from EXEC_MODE register (02h). There are four different modes for each engine, and these modes are described as follows.

Table 5 EXEC MODE register (02h)

Name	Bit	Description
ENG1_EXEC	5:4	Engine 1 program execution **00b**=Hold: Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. **01b**=Step: Execute instruction defined by current Engine 1 PC value, increment PC, and change ENG1_EXEC to 00b (Hold). **10b** = Run: Start at program counter value defined by current Engine 1 PC value. **11b**=Execute instruction defined by current Engine 1 PC value and change ENG1_EXEC to 00b (Hold).
ENG2_EXEC	3:2	Engine 2 program execution **00b**=Hold: Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. **01b**=Step: Execute instruction defined by current Engine 2 PC value, increment PC, and change ENG2_EXEC to 00b (Hold). **10b**=Run: Start at program counter value defined by current Engine 2 PC value. **11b**=Execute instruction defined by current Engine 2 PC value and change ENG2_EXEC to 00b (Hold).
ENG3_EXEC	1:0	Engine 3 program execution **00b**=Hold: Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. **01b**=Step: Execute instruction defined by current engine 3 PC value, increment PC, and change ENG3_EXEC to 00b (Hold). **10b**=Run: Start at program counter value defined by current engine 3 PC value. **11b**=Execute instruction defined by current engine 3 PC value and change ENG3_EXEC to 00b (Hold).

Each engine (1, 2, 3) execution mode can be configured separately. Mode registers are synchronized to a 32kHz clock. Delay between consecutive I²C writes to EXEC MODE register (02h) need to be longer than 200μs (typ.). It should be noted that, when any engine is in load program mode, the exec mode cannot be configured.

EXEC MODE register is allowed to be configured only when all engines are exited load program mode and wait for at least 500μs.

PROGRAM EXECUTION ENGINE OPERATION MODES

Operation modes are defined in register address 03h. Each engine (1, 2, 3) operation mode can be configured separately. Mode registers are synchronized to a 32kHz clock. Delay between consecutive I²C writes to OP_MODE register (03h) need to be longer than 200μs (typ.).

Table 6 Operation Mode Register (OP_MODE (03h))

Name	Bit	Description
ENG1_MODE	5:4	Engine 1 operation mode **00b**=Disabled, reset engine 1 PC **01b**=Load program to SRAM, reset engine 1 PC **10b**=Run program defined by ENG1_EXEC **11b**=Direct control from B PWM I²C register, reset engine 1 PC
ENG2_MODE	3:2	Engine 2 operation mode **00b**=Disabled, reset engine 2 PC **01b**=Load program to SRAM, reset engine 2 PC **10b**=Run program defined by ENG2_EXEC **11b**=Direct control from G PWM I²C register, reset engine 2 PC
ENG3_MODE	1:0	Engine 3 operation mode **00b**=Disabled, reset engine 3 PC **01b**=Load program to SRAM, reset engine 3 PC **10b**=Run program defined by ENG3_EXEC **11b**=Direct control from R PWM I²C register, reset engine 3 PC

Operation Modes

• ***Disabled***

Each channel can be configured to disabled mode. For the current engine mapped LED output brightness will be 0 during this mode. Disabled mode resets respective engine's PC. In this mode, the program execution module of current engine is reset, but the content of SRAM is not affected.

• ***Load program***

AW21104FOR can store 16 commands for each engine (1, 2, 3). Each command consists of 16 bits. Because one register has only 8 bits, one command requires two I²C register addresses. In order to reduce program load time the AW21104FOR supports address auto increment. Register address is incremented after each 8 data bits. The whole program memory can be written in one I²C write sequence. Program memory is defined in the AW21104FOR register table, from address 20h to address 3Fh for engine 1, from address 40h to address 5Fh for engine 2, and from address 60h to address 7Fh for engine 3. In order to access program memory at least one channel operation mode needs to be load program.

SRAM memory writes are allowed only to the channel in load program mode. While one or several engines are in load program mode, PWM and PC values are frozen for the engines which are not in load program mode, and these engines are in paused states, program execution continues when all engines are out of load program mode. PWM and PC values are reset for the engines which are in load program mode.

• ***Run program***

Run program mode executes the commands defined in program memory for respective engine (1, 2, 3). Execution register bits in EXEC_MODE register (02h) define how the program is executed. The program start position can be programmed to Program Counter register (19h, 1Ah, 1Bh). If program counter runs to end (15), next command will be executed from program location 0. Execution registers are synchronized to 32kHz clock. Delay between consecutive I²C writes to OP_MODE register (03h) need to be longer than 200μs (typ.).

Note that entering LOAD program or Direct Control Mode from RUN PROGRAM mode is not allowed. RUN PROGRAM mode only can be switched to DISABLE MODE, and engine execution mode should be in HOLD mode at this time.

• ***Direct control***

In Direct control mode the engine PWM output is controlled by B, G and R PWM I²C registers.

When engine 1 is in Direct control mode, the engine 1 PWM output is controlled by B PWM I²C register (09h).

When engine 2 is in Direct control mode, the engine 2 PWM output is controlled by G PWM I²C register (0Ah).

When engine 3 is in Direct control mode, the engine 3 PWM output is controlled by R PWM I²C register (0Bh).

PROGRAM EXECUTION ENGINE PROGRAM COUNTER (PC)

Program execution engine Program Counter tells the current program code command, which engine is executing. By setting the program counter value before starting the engine execution, user can set the starting point of the program execution.

PC registers are synchronized to 32kHz clock. Delay between consecutive I²C writes to Program Counter (PC) registers (19h, 1Ah, 1Bh) need to be longer than 200μs (typ.).

Table 7 Engine1 PC Register (19h), Engine2 PC Register (1Ah), Engine3 PC Register (1Bh)

Name	Bit	Description
PC	3:0	Program counter value from 0 to 15d

PROGRAM EXECUTION ENGINE PROGRAMMING COMMANDS

The AW21104FOR has three independent programmable engines (1, 2, 3). All engines have own program memory sections for storing LED lighting patterns. Program execution is timed with 32.7kHz clock. This clock can be generated internally or an external 32kHz clock can be connected to the CLK/LEDST pin. Selection of the clock is made with address 05h bits CLK_MODE_SEL. Supported commands are listed in the table below.

Table 8 LED Controller Programming Commands(NOTE3)

<table>
<tr><th>Command</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>RampWait</td><td>0</td><td>Prescale</td><td colspan="6">Step time</td><td>sign</td><td colspan="7">Increment (number of steps)</td></tr>
<tr><td>Set PWM</td><td>0</td><td>1</td><td colspan="6">0</td><td colspan="8">PWM Value</td></tr>
<tr><td>Go to Start</td><td>0</td><td>0</td><td colspan="6">0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Branch</td><td>1</td><td>0</td><td>1</td><td colspan="6">Loop count</td><td colspan="3">x</td><td colspan="4">Step/command number</td></tr>
<tr><td>End</td><td>1</td><td>1</td><td>0</td><td>Int</td><td>Reset</td><td colspan="11">x</td></tr>
<tr><td>Trigger</td><td>1</td><td>1</td><td>1</td><td colspan="3">x</td><td colspan="3">Wait for trigger on engines 1,2,3</td><td colspan="3">x</td><td colspan="3">Send trigger to engines 1,2,3</td><td>x</td></tr>
</table>

NOTE3: X means do not care whether 1 or 0, similar to the following text.

Ramp/Wait

The ramp command generates a PWM ramp starting from current value. At each ramp step the output is incremented by one. Time for one step is defined with Prescale and Step time bits. When Prescale=0, time for one step is (Step time x0.49ms). When Prescale=1, time for one step is (Step time x15.6ms). So it is possible to program very fast and also very slow ramps. Increment value defines how many steps are taken in one command. Number of actual steps is Increment+1. Maximum value is 127d, which corresponds to half of full scale (128 steps). If during ramp command PWM reaches minimum/maximum (0/255) ramp command will be executed to the end and PWM will stay at minimum/maximum. This enables the ramp command to be used as combined ramp and wait command in a single instruction.

The ramp command can be used as wait instruction when increment is zero.

Table 9 Ramp/Wait Command

Ramp/Wait command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	Prescale	Step time						sign	Increment (number of steps)						

Table 10 Ramp/Wait Command Bits

Name	Value(d)	Description
Prescale	0	Divides master clock (32.768 kHz) by 16=2048 Hz, 0.49ms cycle time
	1	Divides master clock (32.768 kHz) by 512=64Hz, 15.6ms cycle time
Step time	1-63	One ramp increment done in (step time) x (clock after prescale) Note: 0 means set PMW command.
Sign	0	Increase PWM output
	1	Decrease PWM output
Increment	0-127	The number of steps is Increment + 1. Note: 0 is a wait instruction

Application Example:

For example if Ramp command=4204h, it means following parameters are used for ramp:

- Prescale =1 //cycle time=15.6ms
- Step time=2 //time=15.6msx2=31.2ms
- Sign =0 //rising ramp
- Increment =4 //5 cycles

If current PWM value is 3, and the first command is as described above, the next command is a ramp with otherwise same the parameters, but with Sign=1 (Command=4284h), the result will be like in the following figure:

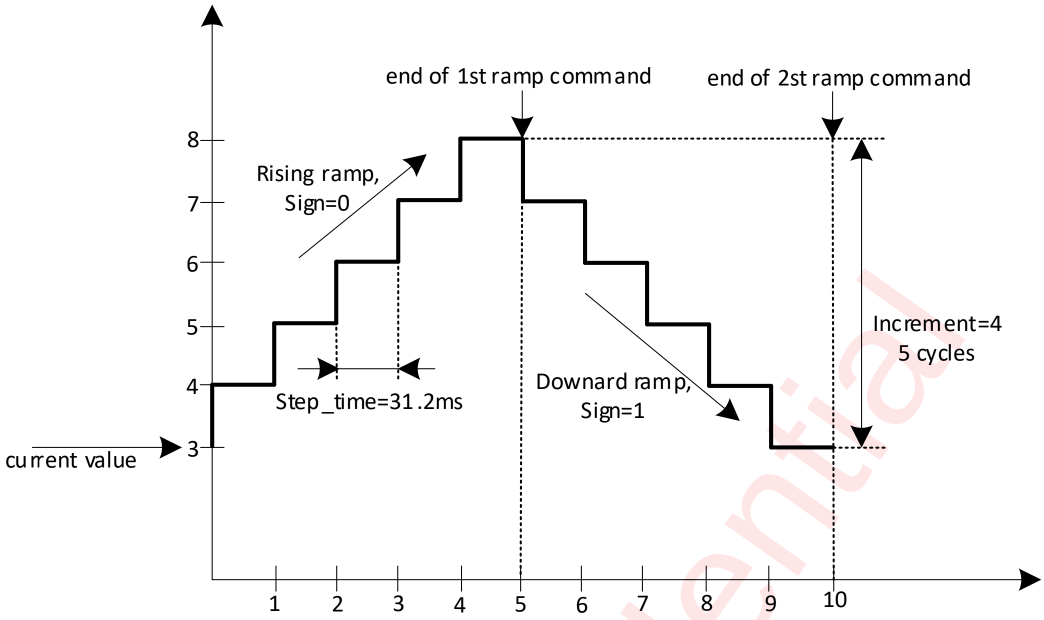


Figure 11 Example of 2 sequential ramp commands

The following table describes condition for entering PS mode when the rampwait commands is executed.

Table 11 rampwait commands and Power Save

Command	Power save condition
Wait	Enables power save, no PWM activity and current command wait time longer than 50ms. If prescale=1 then wait time needs to be longer than 80ms.
Ramp	Enables power save, ramp Command PWM value reaches minimum 0 and current command execution time left more than 50ms. If prescale=1 then time left needs to be more than 80ms.

Set PWM

Set PWM output value from 0 to 255. Command takes sixteen 32kHz clock cycles (488μs).

Table 12 Set PWM command bits

Set PWM command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	1	0						PWM Value							

The following table describes condition for entering PS mode when the set PWM commands is executed.

Table 13 PWM command and Power Save

Command	Power save condition
Set PWM	Enables power save if PWM set to 0 and next command generates at least 50ms wait.

Go-to-Start

Go-to-start command resets the Program Counter register and continues executing program from the 00h location. Command takes sixteen 32kHz clock cycles. Note that default value for all program memory registers is 0000h, which is Go-to-Start command.

Table 14 Go-to-Start Command Bits

Go-to-Start command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Branch

When branch command is executed, the 'Step number' value is loaded to PC, and program execution continues from this location. Looping is done by the number defined in loop count parameter. Nested looping is supported (loop inside loop). The number of nested loops is 8. Command takes sixteen 32kHz clock cycles.

Table 15 Branch Command

Branch command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	0	1	Loop count						x			Step number			

Table 16 Branch Command Bits

Name	Value(d)	Description
loop count	0-63	The number of loops to be done. 0 means infinite loop.
step number	0-15	The step number to be loaded to program counter.

The structure of maximum number nested loops is shown blew.

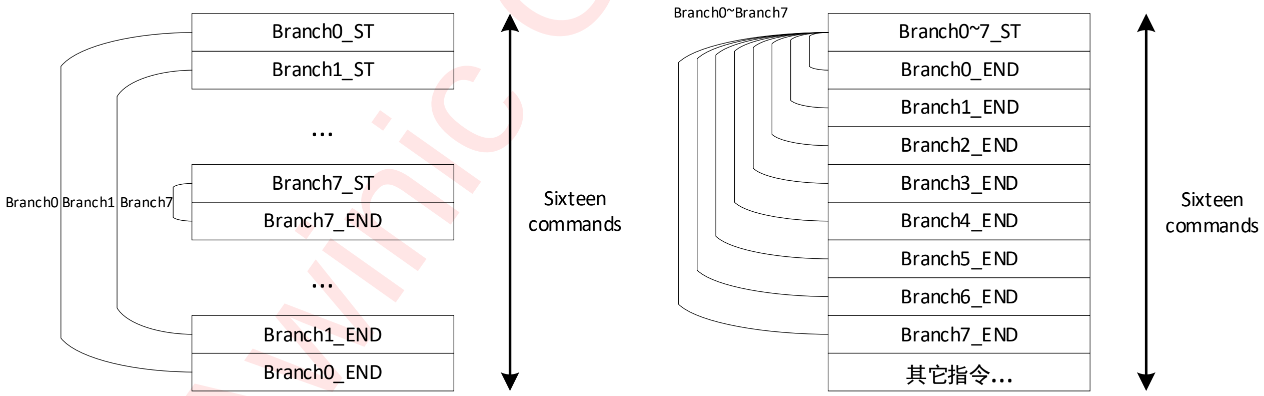


Figure 12 structure of maximum number nested loops

End

End program execution resets the program counter and sets the corresponding EXEC register to 00b (hold). Command takes sixteen 32kHz clock cycles.

Table 17 End Command

End command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	1	0	int	reset						x					

Table 18 End Command Bits

Name	Value	Description
int	0	No interrupt will be sent.
	1	Send interrupt by setting corresponding status register bit high to notify that program has ended. Interrupt can only be cleared by reading interrupt status register 18h.
reset	0	Keep the current PWM value.
	1	Set PWM value to 0

The following table describes condition for entering PS mode when the End commands is executed.

Table 19 End command and Power Save

Command	Power save condition
End	Enables power save, No PWM activity or Reset bit = 1.

Trigger

Wait or send triggers can be used to synchronize operation between different engines. The send-trigger command takes sixteen 32 kHz clock cycles; the wait-for-trigger command takes at least sixteen 32 kHz clock cycles. The receiving engine stores sent triggers. Received triggers are cleared by wait for trigger command if received triggers match to engines defined in the command. Engine waits until all defined triggers have been received.

Table 20 Trigger Command

Trigger command															
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	1	1		x			wait trigger<2:0>			x			send trigger<2:0>		x
						ENG3	ENG2	ENG1				ENG3	ENG2	ENG1	

Table 21 Trigger Command Bits

Name	Value(d)	Description
wait trigger<2:0>	0-7	Wait for trigger for the engine(s) defined. Several triggers can be defined in the same command. Bit 0 is engine 1, bit 1 is engine2, bit 2 is engine 3.
send trigger<2:0>	0-7	Send trigger for the engine(s) defined. Several triggers can be defined in the same command. Bit 0 is engine 1, bit 1 is engine2, bit 2 is engine 3.

The following table describes condition for entering PS mode when the End commands is executed.

Table 22 Trigger Command Bits

Command	Power save condition
Trigger	Enables power save, No PWM activity during wait for trigger command execution.

POWER SAVE MODE

In power save mode analog blocks are disabled to minimize power consumption.

Automatic power save mode is enabled when the PS_EN bit in register address 01h is 1. Almost all analog blocks are powered down in power save, if an external clock is used. However, if an internal clock has been selected, only the LED drivers are disabled during power save since the digital part of the LED controller need to remain active. During program execution the AW21104FOR can enter power-save mode if there is no PWM activity in engine controlled outputs. To prevent short power-save sequences during program execution, the AW21104FOR has a command look-ahead filter. In each instruction cycle every engine commands are analyzed, and if there is sufficient time left with no PWM activity, the device will enter power save. In power save program execution continues uninterruptedly. When a command that requires PWM activity is executed, fast internal startup sequence will be started automatically. The following tables describe commands and conditions that can activate power save. All engines need to meet power-save conditions in order to enable power save.

Table 23 Engine Operation Mode and Power Save

Engine operation mode	Power save condition
00b	Disabled mode enables power save
01b	Load program to SRAM mode prevents power save
10b	Run program mode enables power save if there is no PWM activity and command look-ahead filter condition is met
11b	Direct control mode enables power save if there is no PWM activity

Table 24 Engine Commands and Power Save

Command	Power save condition
Wait	No PWM activity and current command wait time longer than 50ms.If prescale = 1 then wait time needs to be longer than 80ms
Ramp	Ramp Command PWM value reaches minimum 0 and current command execution time left more than 50ms. If prescale =1 then time left needs to be more than 80ms
Trigger	No PWM activity during wait for trigger command execution
End	No PWM activity or Reset bit = 1
Set PWM	Enables power save if PWM set to 0 and next command generates at least 50ms wait
Other commands	No effect to power save

EXTERNAL CLOCK

AW21104FOR has an external clock detector that can detect the presence of an external clock. The external clock detector can detect external clock with frequency above 4kHz, if an external clock is stuck-at-zero or stuck-at-one, or the clock frequency is too low (lower than 4kHz), the clock detector indicates that external clock is not present.

The execution of the program uses an external clock or an internal 32kHz clock. Switching between internal and external clocks using the CLK_MODE_SEL bit of the 05h register, as seen in Table 25. External clock status can be checked with read only bit EXT_CLK_USED in register address 18h, when the automatic selection is enabled (CLK_MODE_SEL=10b), EXT_CLK_USED=1 indicates the presence of an external clock, otherwise it does not present; when the external clock is selected (CLK_MODE_SEL=00b), EXT_CLK_USED=1; in the other cases (CLK_MODE_SEL=01b/11b) EXT_CLK_USED=0.

Table 25 CONFIG0 Register(05h)

Name	Bit	Device Address
CLK_MODE_SEL	1:0	LED Controller clock source 00b = External clock source (CLK) 01b = Internal clock 10b = Automatic selection 11b = Internal clock

THERMAL SHUTDOWN

If the AW21104FOR reaches thermal shutdown temperature (150°C typ.) the device operation is disabled and the device state is in STARTUP mode, until no thermal shutdown event is present. Device will enter Normal mode when temperature drops below 130°C (typ.) degree. Fault is cleared when thermal shutdown disappears.

LOGIC INTERFACE OPERATIONAL DESCRIPTION

The AW21104FOR features a flexible logic interface for connecting to processor and peripheral devices. Communication is done with the I^{2}C-compatible interface, and different logic input/output pins makes it possible to synchronize operation of several devices.

I/O LEVELS

I^{2}C interface, CLK/LEDST. AD0, and AD1 pins input levels are defined by voltage in EN pin. Using the EN pin as a voltage reference for logic inputs simplifies PCB routing and eliminates the need for a dedicated VIO pin. The power supply range of EN is 1.2V~VDD, The following block diagram describes EN pin connections.

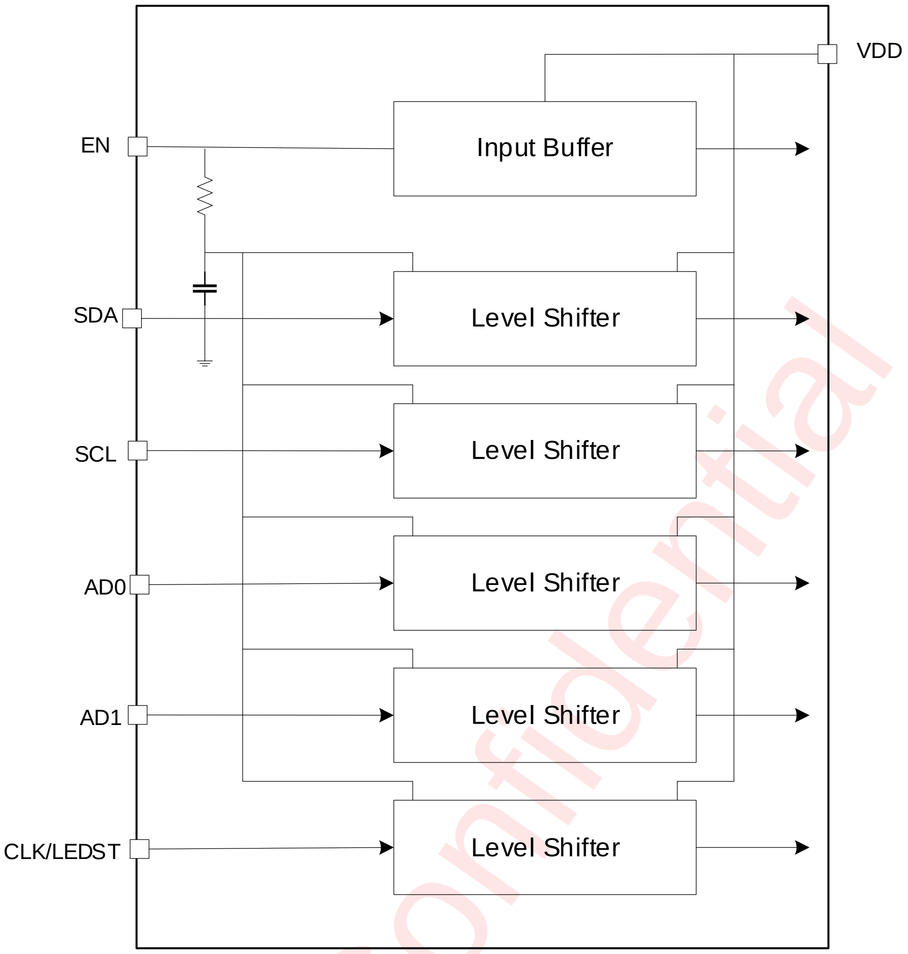


Figure 13 Using EN Pin as Digital I/O Voltage Reference

AD0, AD1 PINS

The AD0 and AD1 pins define the device I²C address. Pins are referenced to EN pin signal level. See I²C device addresses for I²C address definitions.

CLK/LEDST PIN

The CLK/LEDST pin of AW21104FOR is connected to an external 32kHz clock. Engine can use external 32kHz clock to execute commands. Using an external clock can also improve automatic power save mode efficiency, because an internal clock can be switched off automatically when device has entered power-save mode, and an external clock is present. Device can be used without the external clock. If external clock is not used on the application, the CLK/LEDST pin should be connected to GND to prevent floating of this pin and extra current consumption.

I²C INTERFACE

The AW21104FOR supports the I²C protocol. The maximum frequency supported by the I²C is 1MHz. The pull-up resistor for the SDA and SCL can be selected from 1k to 10kΩ. Usually, 4.7kΩ is recommended for 400kHz I²C, 1kΩ is recommended for 1MHz I²C.

DEVICE ADDRESS

Table 26 I²C Device Address Configuration

AD1 Connection	AD0 Connection	Device Address
GND	GND	0x30
GND	VCC	0x31
VCC	GND	0x32
VCC	VCC	0x33

The I²C device address of AW21104FOR depends on the status of pins AD0 and AD1. Connecting pin AD0 or AD1 to GND or VCC will change the device address as showed in table above.

I²C START/STOP

I²C start: SDA changes from high level to low level when SCL is high level.

I²C stop: SDA changes from low level to high level when SCL is high level.

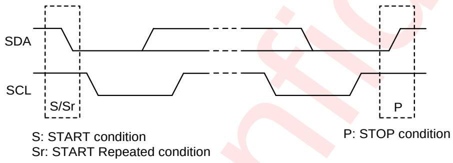


Figure 14 I²C Start/Stop Condition Timing

DATA VALIDATION

When SCL is high level, SDA level must be constant. SDA can be changed only when SCL is low level.

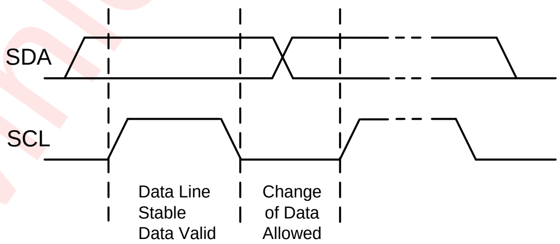


Figure 15 Data Validation Diagram

ACK (ACKNOWLEDGEMENT)

ACK means the successful transfer of I²C bus data. After master sends an 8-bit data, SDA must be released; SDA is pulled to GND by slave device when slave acknowledges.

When master reads, slave device sends 8-bit data, releases the SDA and waits for ACK from master. If ACK is send and I²C stop is not send by master, slave device sends the next data. If ACK is not send by master, slave device stops to send data and waits for I²C stop.

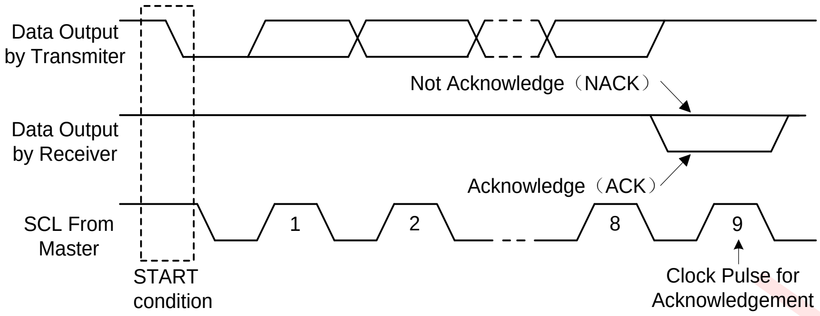


Figure 16 I²C ACK Timing

WRITE CYCLE

One data bit is transferred during each clock pulse. Data is sampled during the high state of the serial clock (SCL). Consequently, throughout the clock's high period, the data should remain stable. Any changes on the SDA line during the high state of the SCL and in the middle of a transaction, aborts the current transaction. New data should be sent during the low SCL state. This protocol allows a single data line to transfer both command/control information and data using the synchronous serial clock.

Each data transaction is composed of a start condition, a number of byte transfers (set by the software) and a stop condition to terminate the transaction. Every byte written to the SDA bus must be 8 bits long and is transferred with the most significant bit first. After each byte, an Acknowledge signal must follow.

In a write process, the following steps should be followed:

a) Master device generates START condition. The "START" signal is generated by lowering the SDA signal while the SCL signal is high.
b) Master device sends slave address (7-bit) and the data direction bit R/W = 0).
c) Slave device sends acknowledge signal if the slave address is correct.
d) Master sends control register address (8-bit)
e) Slave sends acknowledge signal
f) Master sends data byte to be written to the addressed register
g) Slave sends acknowledge signal
h) If master will send further data bytes the control register address will be incremented by one after acknowledge signal (repeat step f and g)
i) Master generates STOP condition to indicate write cycle end

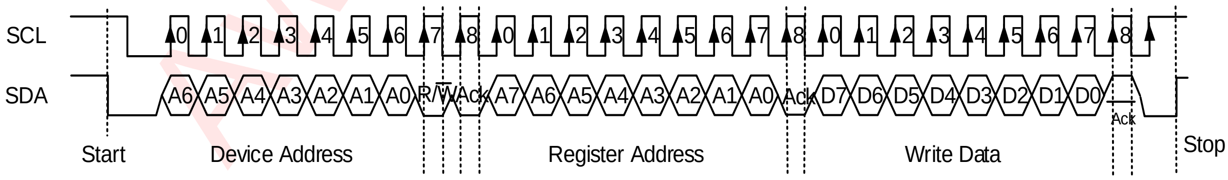


Figure 17 I²C Write Byte Cycle

READ CYCLE

In a read cycle, the following steps should be followed:

a) Master device generates START condition
b) Master device sends slave address (7-bit) and the data direction bit (R/W = 0).
c) Slave device sends acknowledge signal if the slave address is correct.
d) Master sends control register address (8-bit)

e) Slave sends acknowledge signal

f) Master generates STOP condition followed with START condition or REPEAT START condition

g) Master device sends slave address (7-bit) and the data direction bit (R/W = 1).

h) Slave device sends acknowledge signal if the slave address is correct.

i) Slave sends data byte from addressed register.

j) If the master device sends acknowledge signal, the slave device will increase the control register address by one, then send the next data from the new addressed register.

k) If the master device generates STOP condition, the read cycle is ended.

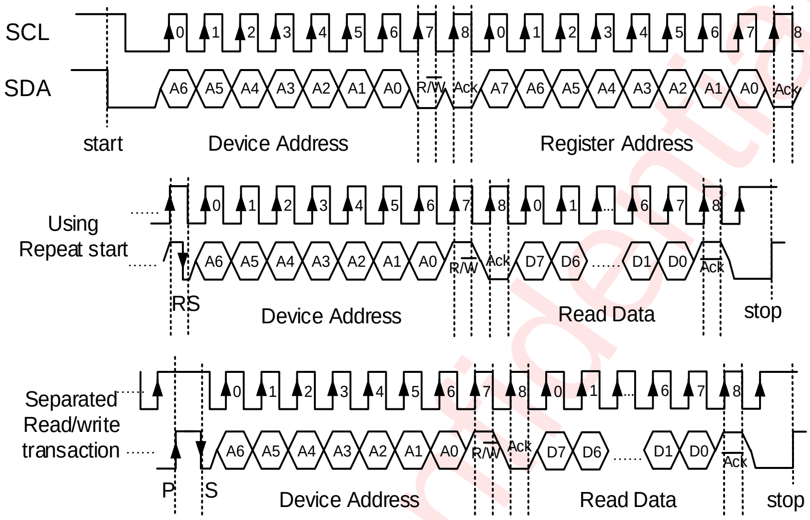


Figure 18 I^{2}C Read Byte Cycle

AUTO-INCREMENT FEATURE

The auto-increment feature allows writing several consecutive registers within one transmission. Every time an 8-bit word is sent to the AW21104FOR, the internal address index counter is incremented by one, and the next register is written.

Register Configuration

Please refer to chapter Programming about SRAM Memory for further information.

REGISTER LIST

<table>
<tr><th>ADDR</th><th>Register</th><th>D7</th><th>D6</th><th>D5</th><th>D4</th><th>D3</th><th>D2</th><th>D1</th><th>D0</th><th>Default</th></tr>
<tr><td>00h</td><td>CHIPID</td><td colspan="8">CHIP_ID</td><td>40h</td></tr>
<tr><td>01h</td><td>ENABLE</td><td colspan="6"></td><td>PS_EN</td><td>CHIP_EN</td><td>00h</td></tr>
<tr><td>02h</td><td>EXEC MODE</td><td colspan="2"></td><td colspan="2">ENG1_EXEC</td><td colspan="2">ENG2_EXEC</td><td colspan="2">ENG3_EXEC</td><td>00h</td></tr>
<tr><td>03h</td><td>OP MODE</td><td colspan="2"></td><td colspan="2">ENG1_MODE</td><td colspan="2">ENG2_MODE</td><td colspan="2">ENG3_MODE</td><td>00h</td></tr>
<tr><td>04h</td><td>LED MAP</td><td colspan="2">W_ENG_SEL</td><td colspan="2">R_ENG_SEL</td><td colspan="2">G_ENG_SEL</td><td colspan="2">B_ENG_SEL</td><td>39h</td></tr>
<tr><td>05h</td><td>CONFIG0</td><td colspan="3"></td><td>CLK32K_OBE</td><td>CLK32K_DOUT_SEL</td><td></td><td colspan="2">CLK_MODE_SEL</td><td>00h</td></tr>
<tr><td>06h</td><td>CONFIG1</td><td colspan="2"></td><td>LOG_EN</td><td></td><td>PINVTE</td><td colspan="3">PWM_FRQ</td><td>04h</td></tr>
<tr><td>07h</td><td>CONFIG2</td><td colspan="3"></td><td>LED_MONITOR_EN</td><td>LED_B_EN</td><td>LED_G_EN</td><td>LED_R_EN</td><td>LED_W_EN</td><td>1Fh</td></tr>
<tr><td>08h</td><td>CONFIG3</td><td>B_LED_ST_EN</td><td>G_LED_ST_EN</td><td>R_LED_ST_EN</td><td>W_LED_ST_EN</td><td>SDA_SR</td><td>LED_SRR</td><td colspan="2">LED_SRF</td><td>F0h</td></tr>
<tr><td>09h</td><td>B PWM</td><td colspan="8">B_PWM</td><td>00h</td></tr>
<tr><td>0Ah</td><td>G PWM</td><td colspan="8">G_PWM</td><td>00h</td></tr>
<tr><td>0Bh</td><td>R PWM</td><td colspan="8">R_PWM</td><td>00h</td></tr>
<tr><td>0Ch</td><td>W PWM</td><td colspan="8">W_PWM</td><td>00h</td></tr>
<tr><td>11h</td><td>B CURRENT</td><td colspan="8">B_CURRENT</td><td>AFh</td></tr>
<tr><td>12h</td><td>G CURRENT</td><td colspan="8">G_CURRENT</td><td>AFh</td></tr>
<tr><td>13h</td><td>R CURRENT</td><td colspan="8">R_CURRENT</td><td>AFh</td></tr>
<tr><td>14h</td><td>W CURRENT</td><td colspan="8">W_CURRENT</td><td>AFh</td></tr>
<tr><td>16h</td><td>RESET</td><td colspan="8">RESET</td><td>00h</td></tr>
<tr><td>17h</td><td>STATUS0</td><td colspan="2"></td><td>BG_OK</td><td>OSC_OK</td><td>LED_B_ST_OUT</td><td>LED_G_ST_OUT</td><td>LED_R_ST_OUT</td><td>LED_W_ST_OUT</td><td>00h</td></tr>
<tr><td>18h</td><td>STATUS1</td><td colspan="3"></td><td>ENG_PS_VALID</td><td>EXT_CLK_USED</td><td>ENG1_INT</td><td>ENG2_INT</td><td>ENG3_INT</td><td>08h</td></tr>
<tr><td>19h</td><td>ENG1 PC</td><td colspan="4"></td><td colspan="4">ENG1_PC</td><td>00h</td></tr>
<tr><td>1Ah</td><td>ENG2 PC</td><td colspan="4"></td><td colspan="4">ENG2_PC</td><td>00h</td></tr>
<tr><td>1Bh</td><td>ENG3 PC</td><td colspan="4"></td><td colspan="4">ENG3_PC</td><td>00h</td></tr>
<tr><td>1Ch</td><td>ENG1 PWM VAL</td><td colspan="8">ENG1_PWM_VAL</td><td>00h</td></tr>
<tr><td>1Dh</td><td>ENG2 PWM VAL</td><td colspan="8">ENG2_PWM_VAL</td><td>00h</td></tr>
<tr><td>1Eh</td><td>ENG3 PWM VAL</td><td colspan="8">ENG3_PWM_VAL</td><td>00h</td></tr>
<tr><td colspan="11">...</td></tr>
<tr><td>20h</td><td>PROG MEM ENG1</td><td colspan="8">CMD1_ENG1[15:8]</td><td>00h</td></tr>
</table>

ADDR	Register	D7	D6	D5	D4	D3	D2	D1	D0	Default
21h	PROG MEM ENG1	CMD1_ENG1[7:0]								00h
...										
3Eh	PROG MEM ENG1	CMD16_ENG1[15:8]								00h
3Fh	PROG MEM ENG1	CMD16_ENG1[7:0]								00h
...										
40h	PROG MEM ENG2	CMD1_ENG2[15:8]								00h
41h	PROG MEM ENG2	CMD1_ENG2[7:0]								00h
...										
5Eh	PROG MEM ENG2	CMD16_ENG2[15:8]								00h
5Fh	PROG MEM ENG2	CMD16_ENG2[7:0]								00h
...										
60h	PROG MEM ENG3	CMD1_ENG3[15:8]								00h
61h	PROG MEM ENG3	CMD1_ENG3[7:0]								00h
...										
7Eh	PROG MEM ENG3	CMD16_ENG3[15:8]								00h
7Fh	PROG MEM ENG3	CMD16_ENG3[7:0]								00h

REGISTER DETAILED DESCRIPTION

CHIPID_CFG: (Address 00h)

Bit	Symbol	R/W	Description	Default
7:0	CHIPID	RO	CHIP ID	0x40

ENABLE: (Address 01h)

Bit	Symbol	R/W	Description	Default
7:2	Reserved	RW	Not used	0
1	PS_EN	RW	Power save mode enable 0: disable 1: enable	0x0
0	CHIP_EN	RW	Master chip enable. Enables device internal startup sequence. Setting EN pin low resets the CHIP_EN state to 0. Allow 500 μs delay after setting chip_en bit to '1' 0: disable 1: enable	0x0

EXEC_MODE: (Address 02h)

Bit	Symbol	R/W	Description	Default
7:6	Reserved	RW	Not used	0
5:4	ENG1_EXEC	RW	Engine 1 program execution. Hold=Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. Step=Execute instruction defined by current engine 1 PC value, increment PC and change ENG1_EXEC to 00b (Hold). Run=Start at program counter value defined by current engine 1 PC value. Execute=Execute instruction defined by current engine 1 PC value and change ENG1_EXEC to 00b (Hold). b00: Hold b01: Step b10: Run b11: Execute	0x0
3:2	ENG2_EXEC	RW	Engine 2 program execution. Hold=Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. Step=Execute instruction defined by current engine 2 PC value, increment PC and change ENG2_EXEC to 00b (Hold). Run=Start at program counter value defined by current engine 2 PC value. Execute=Execute instruction defined by current engine 2 PC value and change ENG2_EXEC to 00b (Hold). b00: Hold b01: Step b10: Run b11: Execute	0x0
1:0	ENG3_EXEC	RW	Engine 3 program execution. Hold=Wait until current command is finished then stop while EXEC mode is hold. PC can be read or written only in this mode. Step=Execute instruction defined by current engine 3 PC value, increment PC and change ENG3_EXEC to 00b (Hold). Run=Start at program counter value defined by current engine 3 PC value. Execute=Execute instruction defined by current engine 3 PC value and change ENG3_EXEC to 00b (Hold). b00: Hold b01: Step b10: Run b11: Execute	0x0

OP_MODE: (Address 03h)

Bit	Symbol	R/W	Description	Default
7:6	Reserved	RW	Not used	0

5:4	ENG1_MODE	RW	Engine 1 operation mode Disabled=Disabled engine. Load program=Load program to SRAM, reset engine 1 PC. Run program=Run program defined by ENG1_EXEC. Direct control=PWM value from I²C. b00: Disabled b01: Load program b10: Run program b11: Direct control Note: Delay need to be longer than 300μs between I²C writes to OP MODE register and I²C writes to EXEC MODE;	0x0
3:2	ENG2_MODE	RW	Engine 2 operation mode Disabled=Disabled engine. Load program=Load program to SRAM, reset engine 2 PC. Run program=Run program defined by ENG2_EXEC. Direct control=PWM value from I²C. b00: Disabled b01: Load program b10: Run program b11: Direct control Note: Delay need to be longer than 300μs between I²C writes to OP MODE register and I²C writes to EXEC MODE;	0x0
1:0	ENG3_MODE	RW	Engine 3 operation mode Disabled=Disabled engine. Load program=Load program to SRAM, reset engine 3 PC. Run program=Run program defined by ENG3_EXEC. Direct control=PWM value from I²C. b00: Disabled b01: Load program b10: Run program b11: Direct control Note: Delay need to be longer than 300μs between I²C writes to OP MODE register and I²C writes to EXEC MODE;	0x0

LED_MAP: (Address 04h)

Bit	Symbol	R/W	Description	Default
7:6	W_ENG_SEL	RW	Selection from where W LED output PWM is controlled b00: I²C register b01: Engine 1 b10: Engine 2 b11: Engine 3	0x0
5:4	R_ENG_SEL	RW	Selection from where R LED output PWM is controlled, b00: I²C register b01: Engine 1 b10: Engine 2 b11: Engine 3	0x3

3:2	G_ENG_SEL	RW	Selection from where G LED output PWM is controlled b00: I²C register b01: Engine 1 b10: Engine 2 b11: Engine 3	0x2
1:0	B_ENG_SEL	RW	Selection from where B LED output PWM is controlled b00: I²C register b01: Engine 1 b10: Engine 2 b11: Engine 3	0x1

CONFIG0: (Address 05h)

Bit	Symbol	R/W	Description	Default
7:5	Reserved	RW	Not used	0
4	CLK32K_OBE	RW	CLK/LEDST pin output enable 0: disable 1: enable	0x0
3	CLK32K_DOUT_SEL	RW	CLK/LEDST pin output select 0: output led status 1: output internal 32kHz clock	0x0
2	Reserved	RW	Not used	0
1:0	CLK_MODE_SEL	RW	LED Controller clock source b00: External clock source (CLK) b01: Internal clock b10: Automatic selection b11: Internal clock	0x0

CONFIG1: (Address 06h)

Bit	Symbol	R/W	Description	Default
7:6	Reserved	RW	Not used	0
5	LOG_EN	RW	Logarithmic PWM adjustment enable bit 0: Linear adjustment 1: Logarithmic adjustment	0x0
4	Reserved	RW	Not used	0
3	PINVTE	RW	PWM inverted output enable 0: disable 1: enable	0x0
2:0	PWM_FRQ	RW	PWM clock frequency 8-bit PWM resolution b000: 4096Hz b001: 2048Hz b010: 1024Hz b011: 512Hz b100: 256Hz b101: 128Hz	0x4

CONFIG2: (Address 07h)

Bit	Symbol	R/W	Description	Default
7:5	Reserved	RW	Not used	0

4	LED_MONITOR_EN	RW	LED driver on or off is controlled by PWM out value when LED monitor enable; if PWM out value is zero, LED driver will be off. 0: disable 1: enable	0x1
3	LED_B_EN	RW	B LED driver enable 0: disable 1: enable	0x1
2	LED_G_EN	RW	G LED driver enable 0: disable 1: enable	0x1
1	LED_R_EN	RW	R LED driver enable 0: disable 1: enable	0x1
0	LED_W_EN	RW	W LED driver enable 0: disable 1: enable	0x1

CONFIG3: (Address 08h)

Bit	Symbol	R/W	Description	Default
7	B_LED_ST_EN	RW	B LED driver switch status output enable 0: disable 1: enable	0x1
6	G_LED_ST_EN	RW	G LED driver switch status output enable 0: disable 1: enable	0x1
5	R_LED_ST_EN	RW	R LED driver switch status output enable 0: disable 1: enable	0x1
4	W_LED_ST_EN	RW	W LED driver switch status output enable 0: disable 1: enable	0x1
3	SDA_SR	RW	SDA slew rate, data on the I^2C bus can be transferred at rate of up to 1Mbit/s when select the high slew rate, data on the I^2C bus can be transferred at rate of up to 400kbit/s when select the low slew rate. 0: high slew rate 1: low slew rate	0x0
2	LED_SRR	RW	R/G/B/W pin rising edge slew rate control 0: 80ns 1: 160ns	0x0
1:0	LED_SRF	RW	R/G/B/W pin falling edge slew rate control b00: 23ns b01: 40ns b10: 60ns b11: 100ns	0x0

B_PWM_CFG: (Address 09h)

Bit	Symbol	R/W	Description	Default
7:0	B_PWM	RW	B LED output PWM value during direct control operation mode	0x0

G_PWM_CFG: (Address 0Ah)

Bit	Symbol	R/W	Description	Default
7:0	G_PWM	RW	G LED output PWM value during direct control operation mode	0x0

R_PWM_CFG: (Address 0Bh)

Bit	Symbol	R/W	Description	Default
7:0	R_PWM	RW	R LED output PWM value during direct control operation mode	0x0

W_PWM_CFG: (Address 0Ch)

Bit	Symbol	R/W	Description	Default
7:0	W_PWM	RW	W LED output PWM value during direct control operation mode	0x0

B_CURRENT_CFG: (Address 11h)

Bit	Symbol	R/W	Description	Default
7:0	B_CURRENT	RW	Current setting b0000 0000: 0.0mA b0000 0001: 0.1mA b0000 0010: 0.2mA ... b1111 1111: 25.5mA	0xAF

G_CURRENT_CFG: (Address 12h)

Bit	Symbol	R/W	Description	Default
7:0	G_CURRENT	RW	Current setting b0000 0000: 0.0mA b0000 0001: 0.1mA b0000 0010: 0.2mA ... b1111 1111: 25.5mA	0xAF

R_CURRENT_CFG: (Address 13h)

Bit	Symbol	R/W	Description	Default
7:0	R_CURRENT	RW	Current setting b0000 0000: 0.0mA b0000 0001: 0.1mA b0000 0010: 0.2mA ... b1111 1111: 25.5mA	0xAF

W_CURRENT_CFG: (Address 14h)

Bit	Symbol	R/W	Description	Default
7:0	W_CURRENT	RW	Current setting b0000 0000: 0.0mA b0000 0001: 0.1mA b0000 0010: 0.2mA ... b1111 1111: 25.5mA	0xAF

RESET_CFG: (Address 16h)

Bit	Symbol	R/W	Description	Default
7:0	RESET	WO	Reset all register values when FFh is written	0x0

STATUS0: (Address 17h)

Bit	Symbol	R/W	Description	Default
7:6	Reserved	RO	Not used	0
5	BG_OK	RO	Bandgap ready 0: disable 1: enable	0x0
4	OSC_OK	RO	OSC ready 0: disable 1: enable	0x0
3	LED_B_ST_OUT	RO	B LED driver state 0: off 1: on	0x0
2	LED_G_ST_OUT	RO	G LED driver state 0: off 1: on	0x0
1	LED_R_ST_OUT	RO	R LED driver state 0: off 1: on	0x0
0	LED_W_ST_OUT	RO	W LED driver state 0: off 1: on	0x0

STATUS1: (Address 18h)

Bit	Symbol	R/W	Description	Default
7:5	Reserved	RO	Not used	0
4	ENG_PS_VALID	RO	Engine enter power save mode state 0: not valid 1: valid	0x0
3	EXT_CLK_USED	RO	External clock state 0: Internal clock used 1: External 32kHz clock used	0x1
2	ENG1_INT	RO	Interrupt from engine 1	0x0
1	ENG2_INT	RO	Interrupt from engine 2	0x0
0	ENG3_INT	RO	Interrupt from engine 3	0x0

ENG1_PC_CFG: (Address 19h)

Bit	Symbol	R/W	Description	Default
7:4	Reserved	RO	Not used	0
3:0	ENG1_PC	RO	Engine 1 program counter value	0x0

ENG2_PC_CFG: (Address 1Ah)

Bit	Symbol	R/W	Description	Default
7:4	Reserved	RO	Not used	0

3:0	ENG2_PC	RO	Engine 2 program counter value	0x0

ENG3_PC_CFG: (Address 1Bh)

Bit	Symbol	R/W	Description	Default
7:4	Reserved	RO	Not used	0
3:0	ENG3_PC	RO	Engine 3 program counter value	0x0

ENG1_PWM_VAL_CFG: (Address 1Ch)

Bit	Symbol	R/W	Description	Default
7:0	ENG1_PWM_VAL	RO	PWM value output by engine 1	0x0

ENG2_PWM_VAL_CFG: (Address 1Dh)

Bit	Symbol	R/W	Description	Default
7:0	ENG2_PWM_VAL	RO	PWM value output by engine 2	0x0

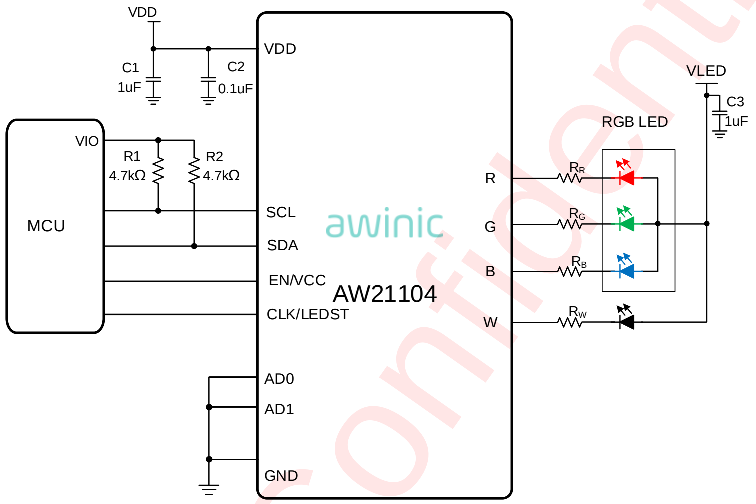
ENG3_PWM_VAL_CFG: (Address 1Eh)

Bit	Symbol	R/W	Description	Default
7:0	ENG3_PWM_VAL	RO	PWM value output by engine 3	0x0

Application Information

The AW21104FOR is a 4-channel LED driver. The device has a program memory for creating variety of lighting sequences. When the program memory has been loaded, the AW21104FOR can operate independently without processor control. An internal of external 32-kHz clock is required. If multiple AW21104FOR devices are used to sequence multiple RGB LEDs, then the external 32-kHz clock input is required. The AD0 and AD1 pins can be used to allow unique sequencing of up to four AW21104FOR devices on the same I2C bus. The four LED current drivers can be configured up to 25.5mA LED current each and are tolerant up to 5.5V LED supply voltage.

Typical Application



Note: The resistor R_{LED} is only thermal reduction, and it is determined by V_{LED}, VF of LED, $V_{DROPOUT}$ of LEDx and I_{LED}.
$R_{LED}=(V_{LED}-VF-V_{DROPOUT})/I_{LED}$.

Figure 19 AW21104FOR Application Circuit

Power Supply Recommendations

The AW21104FOR can be powered by a power source or battery below 5.5V. The resistance of the input supply rail should be low enough so that the input current transient does not cause too high drop at the AW21104FOR VDD pin. If the LED is powered by another power source, a ceramic bypass capacitor needs to be added to the LED power line.

PCB Layout Consideration

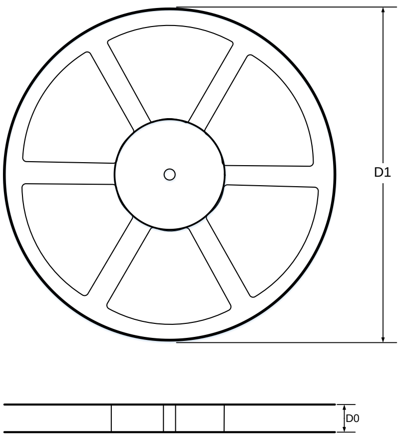
To obtain the good thermal performance, PCB layout should be considered carefully. Here are some guidelines:

1. The C1 and C2 should be placed as close to the chip as possible.
2. If the LED is powered by another power source, the C3 should be placed as close to the LED as possible.

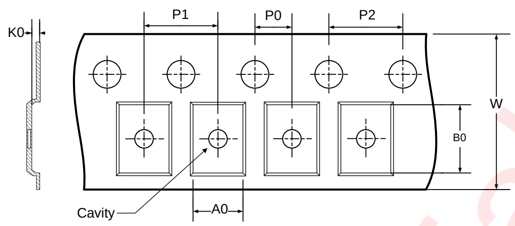
Tape And Reel Information

AW21104FOR

REEL DIMENSIONS

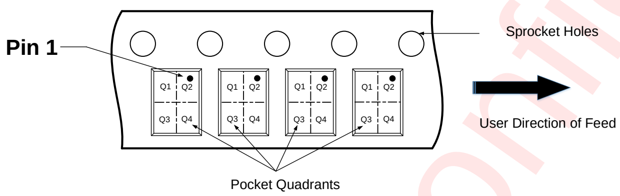


TAPE DIMENSIONS



A0：Dimension designed to accommodate the component width
B0：Dimension designed to accommodate the component length
K0：Dimension designed to accommodate the component thickness
W：Overall width of the carrier tape
P0：Pitch between successive cavity centers and sprocket hole
P1：Pitch between successive cavity centers
P2：Pitch between sprocket hole
D1：Reel Diameter
D0：Reel Width

QUADRANT ASSIGNMENTS FOR PIN 1 ORIENTATION IN TAPE



Note：The above picture is for reference only. Please refer to the value in the table below for the actual size

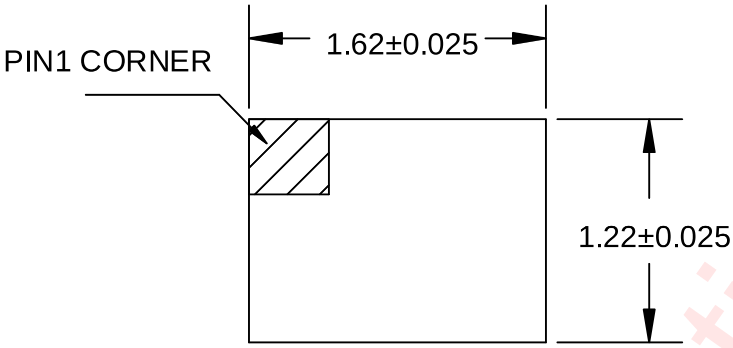
DIMENSIONS AND PIN1 ORIENTATION

D1 (mm)	D0 (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P0 (mm)	P1 (mm)	P2 (mm)	W (mm)	Pin1 Quadrant
179.0	9.00	1.36	1.80	0.74	2.00	4.00	4.00	8.00	Q2

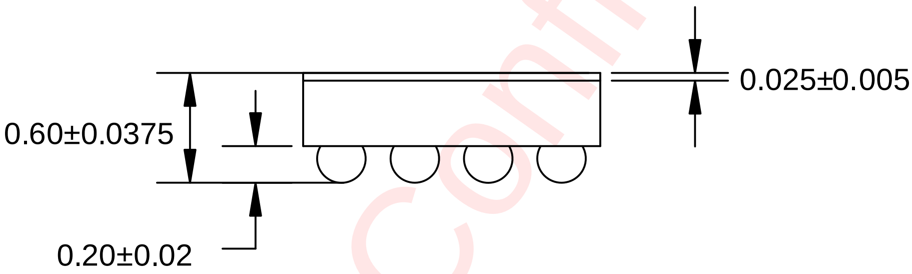
All dimensions are nominal

Package Description

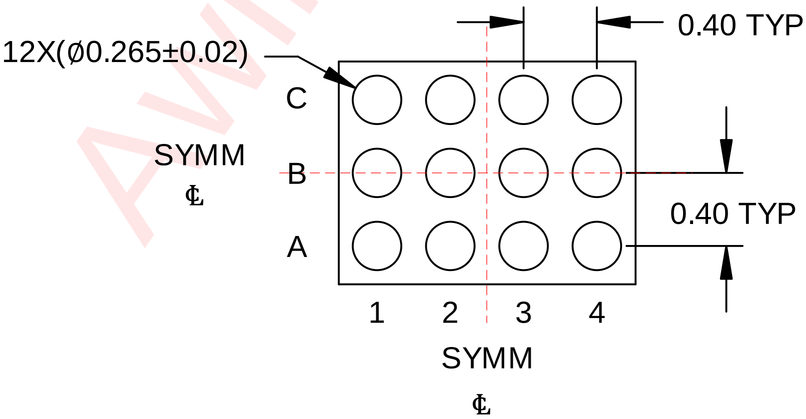
AW21104FOR



Top View



Side View

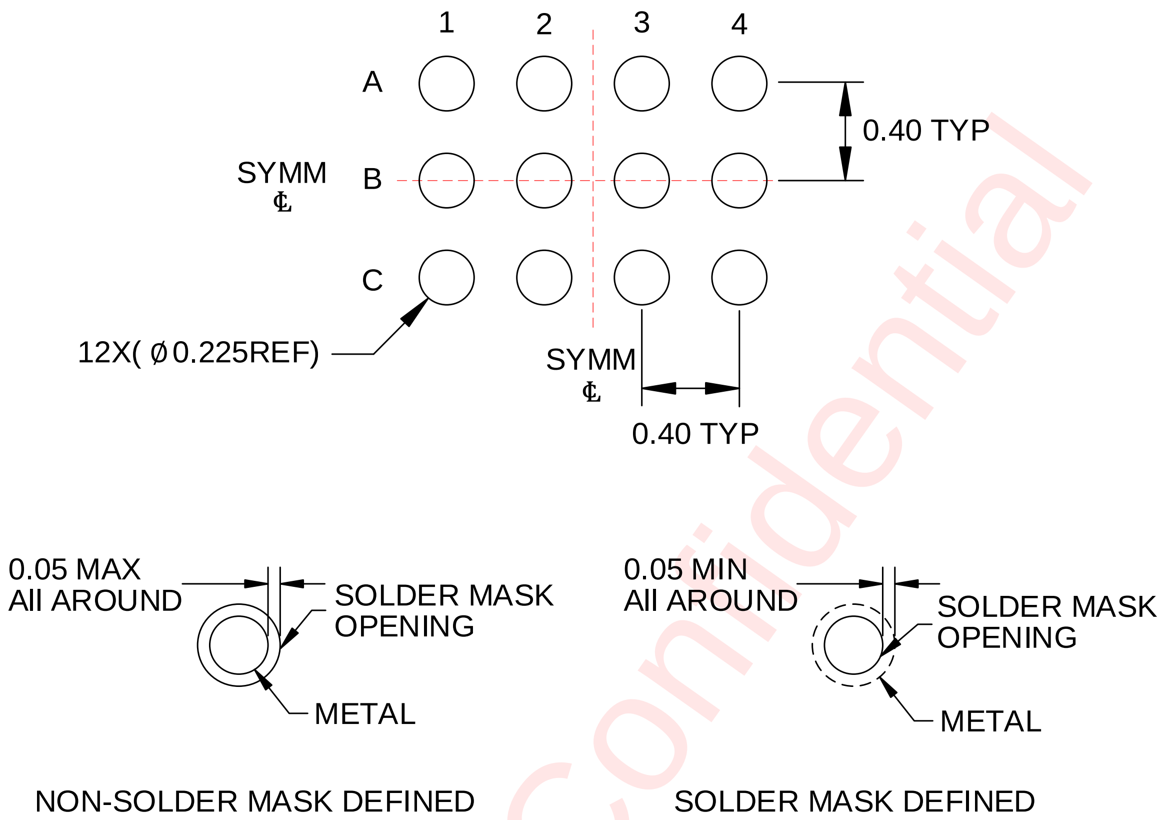


Bottom View

Unit：mm

Land Pattern Data

AW21104FOR



Unit：mm

Revision History

Version	Date	Change Record
V1.0	Jun. 2024	Officially released
V1.1	Sept. 2024	1. Update Current accuracy in Features.(P1) 2. Update I_{MATCH}, POR_{HYS}, V_{IL} in Electrical Characteristics.(P5, P6) 3. Update Register List.(P26)
V1.2	Feb. 2025	1. Update Electrical Characteristics, add min and max of some Parameters. (P5-6) 2. Update power on timing. (P8) 3. Add description of LED status indication. (P13) 4. Update Tape And Reel Information, add tolerance of all dimensions. (P36)
V1.3	Jul. 2025	1. Modify description (P13, 19, 20)
V1.4	Jul. 2025	1. Add new package QNR(P1-3, P37-42)
V1.5	Dec. 2025	1. Delete package QNR(P1-3, P37-42) 2. The pin name of CLK is updated to CLK/LEDST pin(P1-2) 3. Add Led LED Status Indication Diagram(P13) 4. Update branch Command (P16)

Disclaimer

All trademarks are the property of their respective owners. Information in this document is believed to be accurate and reliable. However, Shanghai AWINIC Technology Co., Ltd (AWINIC Technology) does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

AWINIC Technology reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. Customers shall obtain the latest relevant information before placing orders and shall verify that such information is current and complete. This document supersedes and replaces all information supplied prior to the publication hereof.

AWINIC Technology products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of an AWINIC Technology product can reasonably be expected to result in personal injury, death or severe property or environmental damage. AWINIC Technology accepts no liability for inclusion and/or use of AWINIC Technology products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications that are described herein for any of these products are for illustrative purposes only. AWINIC Technology makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

All products are sold subject to the general terms and conditions of commercial sale supplied at the time of order acknowledgement.

Nothing in this document may be interpreted or construed as an offer to sell products that is open for acceptance or the grant, conveyance or implication of any license under any copyrights, patents or other industrial or intellectual property rights.

Reproduction of AWINIC information in AWINIC data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. AWINIC is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of AWINIC components or services with statements different from or beyond the parameters stated by AWINIC for that component or service voids all express and any implied warranties for the associated AWINIC component or service and is an unfair and deceptive business practice. AWINIC is not responsible or liable for any such statements.