Datasheet for LENZE BLE5.0 + IEEE802.15.4 Multi-Standard WirelessSoCST17H58

DS-ST17H58

Ver1.0.1

2019/1/6

Keyword:
Bluetooth Smart; BLE5.0 Mesh; 6LoWPAN ; Thread;Zigbee; RF4CE;HomeKit; 2.4GHz; Feature s; Package; Pin layout; Memory;MCU; Working modes; Wakeup sources; RF Transceiver; Baseband; Clock; Timers; Interrupt; Interface; PWM; Audio; QDEC; ADC; PGA;Comparator; AES; Electrical specification

Brief:
This datasheet is dedicated for Lenze BLE5.0 + IEEE802.15.4 multi-standard SoC ST17H58. In this datasheet, key features, working mode, main modules, electrical specification and application of the ST17H58 are introduced.

LENZE TECHNOLOGY

Published by
Lenze Technology

7th Floor,Ching Po Building,Shenzhen Special Zone Press Group, Futian District, Shenzhen, China

© Lenze Technology
All Right Reserved

Legal Disclaimer

Lenze Technology reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Lenze Technology disclaims any and all liability for any errors, inaccuracies or incompleteness contained herein or in any other disclosure relating to any product.

Lenze Technology does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others

The products shown herein are not designed for use in medical, life-saving, or life-sustaining applications. Customers using or selling Lenze Technology products not expressly indicated for use in such applications do so entirely at their own risk and agree to fully indemnify Lenze Technology for any damages arising or resulting from such use or sale.

Information:

For further information on the technology, product and business term, please contact Lenze Technology Company (www.lenzetech.com).

For sales or technical support, please send email to the address of:

info@lenzetech.com

Revision History

Version	Major Changes	Date	Author
0.8.0	Preliminary release	2018/3	Y.C.Q., S.G.J., T.J.Q., J.J.W., G.X.W., L.Y., Z.X.Y., J.H.P., X.X., S.S., X.S.J., Cynthia
0.8.1	Updated ordering information.	2018/3	X.S.J., Cynthia
0.8.2	Updated section 7.3.1 Communication protocol.	2018/6	S.G.J., Cynthia
1.0.1	Modified some editorial errors.	2019/1	Cynthia

1 Table of contents

1 Overview 9
1.1 Block diagram 9
1.2 Key features 11
1.2.1 General features 11
1.2.2 RF Features 12
1.2.3 Features of power management module 12
1.2.4 USB features 13
1.2.5 Flash features 13
1.2.6 RF4CE features 13
1.2.7 Zigbee features 13
1.2.8 6LowPAN and Thread features 14
1.2.9 BLE Mesh features 14
1.2.10 Concurrent mode feature 14
1.2.11 HomeKit features 15
1.3 Typical applications 15
1.4 Ordering information 15
1.5 Package 16
1.6 Pin layout 18
1.6.1 Pin layout for ST17H58 18
1.6.2 Notes 20
2 Memory and MCU 21
2.1 Memory 21
2.1.1 SRAM/Register 21
2.1.2 Flash 24
2.2 Firmware encryption 24
2.3 MCU 24
2.4 Working modes 24
2.5 Reset 26
2.6 Power Management 27
2.6.1 Power-On-Reset (POR) and Brown-out detect 27
2.6.2 Working mode switch 29
2.6.3 LDO and DCDC 31
2.7 Wakeup sources 31
2.7.1 Wakeup source - USB 31
2.7.2 Wakeup source – 32K timer 31
2.7.3 Wakeup source – low power comparator 31
2.7.4 Wakeup source – IO 31
2.7.5 Register table 32
3 BLE/802.15.4/2.4G RF Transceiver 35
3.1 Block diagram 35
3.2 Function description 35
3.2.1 Turn on/off 35
3.2.2 Air interface data rate and RF channel frequency 35
3.3 Baseband 36
3.3.1 Packet format 36
3.3.2 BLE location function 37
3.3.3 RSSI and frequency offset 37
4 Clock 38

4.1 Clock sources 38
4.2 System clock 39
4.3 Module clock 39
4.3.1 System Timer clock 39
4.3.2 USB clock 39
4.3.3 I2S clock 39
4.3.4 DMIC clock 39
4.4 Register table 39
5 Timers 41
5.1 Timer0~Timer2 41
5.1.1 Register table 41
5.1.2 Mode0 (System Clock Mode) 42
5.1.3 Mode1 (GPIO Trigger Mode) 42
5.1.4 Mode2 (GPIO Pulse Width Mode) 43
5.1.5 Mode3 (Tick Mode) 44
5.1.6 Watchdog 44
5.2 32K LTIMER 46
5.3 System Timer 46
6 Interrupt System 47
6.1 Interrupt structure 47
6.2 Register configuration 47
6.2.1 Enable/Mask interrupt sources 48
6.2.2 Interrupt mode and priority 48
6.2.3 Interrupt source flag 48
7 Interface 49
7.1 GPIO 49
7.1.1 Basic configuration 49
7.1.1.1 Multiplexed functions 49
7.1.1.2 Drive strength 50
7.1.1.3 GPIO lookup table 50
7.1.2 Connection relationship between GPIO and related modules 52
7.1.3 Pull-up/Pull-down resistor 56
7.2 SWM and SWS 57
7.3 I2C 58
7.3.1 Communication protocol 58
7.3.2 Register table 58
7.3.3 I2C Slave mode 59
7.3.3.1 DMA mode 60
7.3.3.2 Mapping mode 60
7.3.4 I2C Master mode 60
7.3.4.1 I2C Master Write transfer 60
7.3.4.2 I2C Master Read transfer 61
7.3.5 I2C and SPI Usage 61
7.4 SPI 61
7.4.1 Register table 61
7.4.2 SPI Master mode 62
7.4.3 SPI Slave mode 64
7.4.4 I2C and SPI Usage 64
7.5 UART 64

7.6 USB 67
8 PWM 68
8.1 Register table 68
8.2 Enable PWM 72
8.3 Set PWM clock 72
8.4 PWM waveform, polarity and output inversion 72
8.4.1 PWM waveform 72
8.4.2 Invert PWM output 74
8.4.3 Polarity for signal frame 74
8.5 PWM mode 74
8.5.1 Select PWM mode 74
8.5.2 Continuous mode 74
8.5.3 Counting mode 75
8.5.4 IR mode 75
8.5.5 IR FIFO mode 76
8.6 PWM interrupt 79
9 Audio 80
9.1 Audio input path 80
9.1.1 AMIC input 80
9.1.2 DMIC input 81
9.1.3 I2S input 81
9.1.4 USB Host input 81
9.1.5 DFIFO 81
9.2 Audio input processing 82
9.2.1 Decimation filter 82
9.2.2 LPF 82
9.2.3 Down-sample 84
9.2.4 HPF 84
9.2.5 ALC 84
9.2.5.1 Auto regulate in analog mode 84
9.2.5.2 Manual regulate in analog mode 85
9.2.5.3 Auto regulate in digital mode 85
9.2.5.4 Manual regulate in digital mode 85
9.2.6 Register configuration 86
9.3 Audio output path 89
9.3.1 Rate Matching 90
9.3.2 SDM 91
9.3.3 Register configuration 91
10 Quadrature Decoder 95
10.1 Input pin selection 95
10.2 Common mode and double accuracy mode 95
10.3 Read real time counting value 97
10.4 QDEC interrupt 98
10.5 QDEC reset 98
10.6 Other configuration 98
10.7 Register table 99
11 SAR ADC 99
11.1 Power on/down 99
11.2 ADC clock 99

11.3 ADC control in auto mode 99
11.3.1 Set max state and enable channel 99
11.3.2 “Set” state 99
11.3.3 “Capture” state 100
11.3.4 Usage cases 101
11.3.4.1 Case 1: 4-channel sampling for random number, stereo audio and Misc 101
11.3.4.2 Case 2: 3-channel sampling for random number, mono audio and Misc 101
11.3.4.3 Case 3: 3-channel sampling for stereo audio and Misc 102
11.3.4.4 Case 4: 3-channel sampling for random number and stereo audio 102
11.3.4.5 Case 5: 2-channel sampling for mono audio and Misc 102
11.3.4.6 Case 6: 2-channel sampling for random number and mono audio 103
11.3.4.7 Case 7: 2-channel sampling for random number and Misc 103
11.3.4.8 Case 8: 2-channel sampling for stereo audio 103
11.3.4.9 Case 9: 1-channel sampling for mono audio 103
11.3.4.10 Case 10: 1-channel sampling for Misc 104
11.3.4.11 Case 11: 1-channel sampling for RNG 104
11.4 Register table 104
12 PGA 110
12.1 Power on/down 111
12.2 Select input channel 111
12.3 Adjust gain 111
12.4 Register table 112
13 Temperature Sensor 113
14 Low Power Comparator 115
14.1 Power on/down 115
14.2 Select input channel 115
14.3 Select mode and input channel for reference 115
14.4 Select scaling coefficient 116
14.5 Low power comparator output 116
14.6 Register table 116
15 Continuous Comparator 118
15.1 Function overview 118
15.2 Differential mode 119
15.3 Single-end mode 120
15.4 Register table 121
16 AES 123
16.1 RISC mode 123
16.2 DMA mode 123
16.3 AES-CCM 123
16.4 Register table 124
17 Key Electrical Specifications 125
17.1 Absolute maximum ratings 125
17.2 Recommended operating condition 125
17.3 DC characteristics 125
17.4 AC characteristics 126

2 Table of Figures

Figure 1- 1 Block diagram of the system 10
Figure 1- 2 Package dimension for the ST17H58 (Unit: mm) 17
Figure 2- 1 Physical memory map 21
Figure 2- 2 Register space 22
Figure 2- 3 Control logic for power up/down 26
Figure 2- 4 Initial Power-up sequence 27
Figure 2- 5 Power-down sequence 28
Figure 2- 6 Wakeup sources 31
Figure 3- 1 Block diagram of RF transceiver 34
Figure 4- 1 Block diagram of clock 37
Figure 7- 1 Logic relationship between GPIO and related modules 53
Figure 7- 2 I2C timing chart 57
Figure 7- 3 Byte consisted of slave address and R/W flag bit 58
Figure 7- 4 Read format in DMA mode 59
Figure 7- 5 Write format in DMA mode 59
Figure 7- 6 Read format in Mapping mode 60
Figure 7- 7 Write format in Mapping mode 60
Figure 7- 8 SPI write/read command format 63
Figure 7- 9 UART communication 64
Figure 8- 1 PWM output waveform chart 73
Figure 8-2 Continuous mode 74
Figure 8-3 Counting mode (n=0) 74
Figure 8-4 IR mode (n=0) 75
Figure 8- 5 Waveform structure in IR FIFO mode 76
Figure 9- 1 Audio input path 80
Figure 9- 2 Audio input processing 82
Figure 9- 3 Audio output path 90
Figure 9- 4 Linear interpolation 90
Figure 9- 5 Delay interpolation 90
Figure 9- 6 Block diagram of SDM 91
Figure 10- 1 Common mode 95
Figure 10- 2 Double accuracy mode 96
Figure 10- 3 Read real time counting value 97
Figure 10- 4 Shuttle mode 97
Figure 12- 1Block diagram of PGA 110
Figure 13- 1 Block diagram of temperature sensor 113
Figure 14- 1 Block diagram of low power comparator 115
Figure 15- 1 Block diagram of continuous comparator 118
Figure 15- 2 Continuous comparator in differential mode 119
Figure 15- 3 Continuous comparator in single-end mode 120

3 Table of Tables

Table 1- 1 Ordering information of the ST17H58 15
Table 1- 2 Pin functions for ST17H58 18
Table 2- 1 Working modes 24

Table 2- 2 Register configuration for software reset 25
Table 2- 3 Analog register to control delay counters 27
Table 2- 4 Characteristics of Initial Power-up/ Power-down sequence 28
Table 2- 5 3.3V analog registers for module power up/down control 29
Table 2- 6 Analog registers for Wakeup 32
Table 2- 7 Digital register for Wakeup 33
Table 3- 1 Packet Format in standard 1Mbps BLE mode.....35
Table 3- 2 Packet format in standard 2Mbps BLE mode 35
Table 3- 3 Packet format in standard 500Kbps/250Kbps BLE mode 35
Table 3- 4 Packet format in 802.15.4 mode 35
Table 3- 5 Packet format in Proprietary mode.....36
Table 4- 1 Register table related to clock.....39
Table 5- 1 Register configuration for Timer0~Timer2 41
Table 5- 2 Register table for System Timer 45
Table 6- 1 Register table for Interrupt system 46
Table 7- 1 GPIO lookup table 49
Table 7- 2 GPIO lookup table2 54
Table 7- 3 Analog registers for pull-up/pull-down resistor control.....55
Table 7- 4 Register configuration for I2C 57
Table 7- 5 Register configuration for SPI.....61
Table 7- 6 SPI mode 62
Table 7- 7 Register configuration for UART.....64
Table 8- 1 Register table for PWM 67
Table 9- 1 Audio data flow direction.....80
Table 9- 2 Register configuration related to audio input processing 85
Table 9- 3 Register configuration related to audio output path 92
Table 10- 1 Input pin selection 94
Table 10- 2 Register table for QDEC.....98
Table 11- 1 Register table related to SAR ADC.....104
Table 12- 1 Analog register table related to PGA.....112
Table 13- 1 Analog register for temperature sensor.....113
Table 14- 1 Analog register table related to low power comparator.....116
Table 15- 1 Analog register table related to continuous comparator.....121
Table 16- 1 Register table related to AES.....124
Table 17- 1 Absolute Maximum Ratings 125
Table 17- 2 Recommended operation condition 125
Table 17- 3 DC characteristics (T=25℃).....125
Table 17- 4 AC Characteristics 126

1 Overview

The ST17H58 is Lenze-developedBluetoothLE + IEEE802.15.4 multi-standard wirelessSoC solution with internal Flash and audio support, which combines the features and functions needed for all 2.4GHz IoT standards into a single SoC.It's completely RoHS-compliant and 100% lead (Pb) -free.

The ST17H58 combines the radio frequency (RF), digital processing, protocols stack software and profiles for multiple standards into a single SoC. The chip supports standards and industrial alliance specifications including Bluetooth Low Energy (up to Bluetooth LE5.0), BLE Mesh, 6LoWPAN, Thread, Zigbee, RF4CE, HomeKit, ANT and 2.4GHz proprietary standard. The ST17H58'sembedded 512KB FLASH enables dynamic stack and profile configuration, and the final end product functionality is configurable via software, providing ultimate flexibility. The ST17H58 also has hardware OTA upgrades support and multiple boot switching, allowing convenient product feature roll outs and upgrades.

The ST17H58 supports concurrent multi-standards. For some use cases, the ST17H58 can "concurrently" run two standards, for example, stacks such as BLEand802.15.4 can run concurrently with one application state but dual radio communication channels for interacting with different devices.The end product working in this mode can maintain active Bluetooth Smart connections to smart phones or other BLE devices while control and communicate with 802.15.4 or other 2.4Ghz devices at the same time. In this case, it's compatible with Bluetooth standard, supports BLE specification up to version 5.0, allows easy connectivity with Bluetooth Smart Ready mobile phones, tablets, laptops, which supports BLE slave and master mode operation, including broadcast, encryption, connection updates, and channel map updates. At the same time, it also supports IEEE 802.15.4 standard and Zigbee-compliant platform, and is perfect for creating interoperable solution for use within the home combined with leading Zigbee/RF4CE software stack. This feature enables products to bridge the smartphone and home automation world with a single chip and no requirement for an external hub.

The ST17H58 integrates hardware acceleration to support the complicated security operations required by HomeKit, Thread and other standards without the requirement for an external DSP, thereby significantly reducing the producteBOM.

The ST17H58 supports single or dual analog microphones, digital microphone, and stereo audio output with enhanced voice performance for voice search and other such applications. The ST17H58 also includes a full range of on-chip peripherals for interfacing with external components such as LEDs, sensors, touch controllers, keyboards, and motors. This makes it an ideal single-chip solution for IoT (Internet of Things) and HID (Human Interface Devices) applicationsuch as wearable devices, smart lighting, smart home devices, advanced remote controls, and wireless toys.

1.1 Block diagram

The ST17H58 is designed to offer extremely high integration, ultra-low power application capabilities.The system's block diagram is as shown in Figure 1-1.

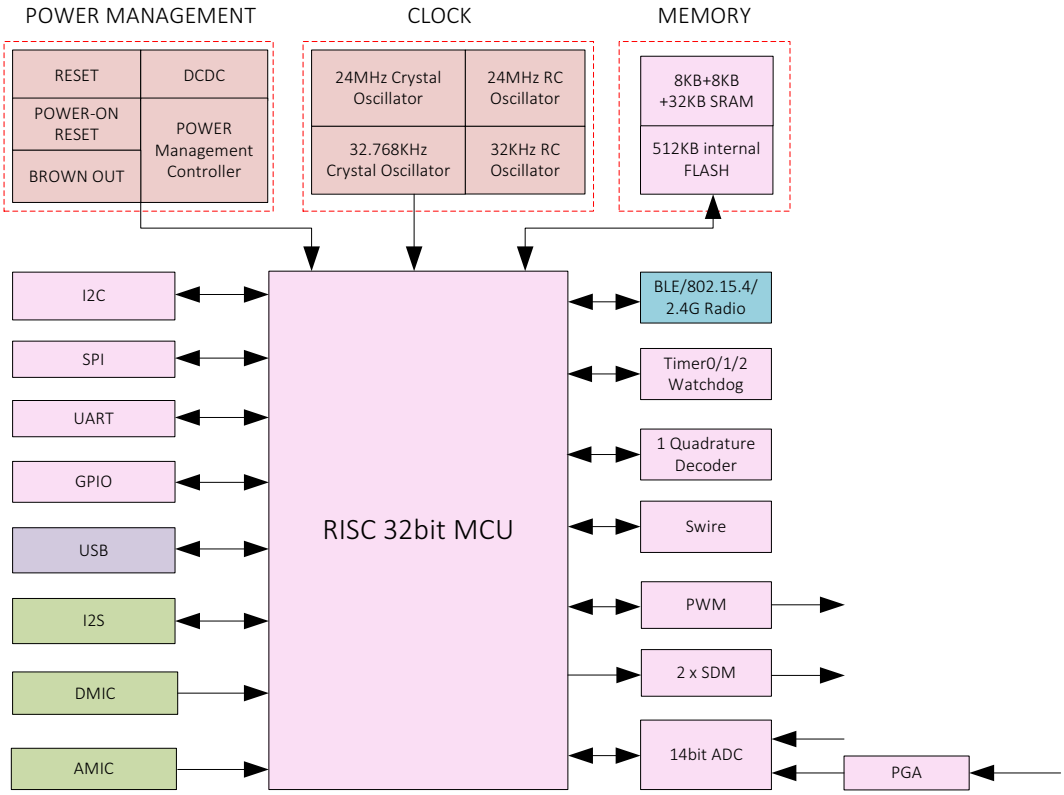


Figure 1- 1 Block diagram of the system

*Note:

1) Modules marked with different colors belong to different power domains.Power state of each power domain can be controlled independent of other power domains, for example, the audio module (including I2S, DMIC, AMIC) can be independently powered on or powered downirrespective of other modules such as power management module, clock, and etc.
2) The BLE/802.15.4/2.4G Radio, USB and Audio (I2S, DMIC, AMIC) are powered down by default.
3) The power management module and clock should be always powered on, even in deep sleep.
4) In deep sleep, except for the power management and clock,allother modules should be powered down.

The ST17H58 integrates strong 32-bit MCU, BLE/802.15.4/2.4G Radio, 48KB (8K +8K+32K) SRAM, 512KB internal Flash, 14bit ADC with PGA, analog and digital microphone input, stereo audio output, 6-channel PWM(1-channel IR/IR FIFO), one quadrature decoder (QDEC), abundant and flexible GPIO interfaces, and nearly all the peripherals needed for IoT (Internet of Things) and HID (Human Interface Devices)a pplication development (e.g. Bluetooth Low Energy and Zigbee/IEEE 802.15.4/RF4CE) .The ST17H58 also includes multi-stage power management design allowing ultra-low power operation and making it the ideal candidate for wearable and power-constraint applications.

With the high integration level of ST17H58, few external components are needed to satisfy customers' ultra-low cost requirements.

1.2 Key features

1.2.1 General features

General features are as follows:

1) 3-byte or 7-byte Unique Chip ID (UID).
2) Embedded 32-bit high performance MCU with clock up to 48MHz.
3) Program memory: internal 512KB Flash.
4) Data memory: 48KBon-chip SRAM, including two independent 8KB SRAMs with retention in deep sleep, and one 32KB SRAM without retention in deep sleep.
5) 24MHz & 32.768KHz Crystal and 32KHz/24MHz embedded RC oscillator.
6) A rich set of I/Os:
   - ✧ Up to 32/17/10GPIOs depending on package option. All digital IOs can be used as GPIOS.
   - ✧ DMIC (Digital Mic).
   - ✧ AMIC (Analog Mic).
   - ✧ I2S.
   - ✧ StereoAudio output.
   - ✧ SPI.
   - ✧ I2C.
   - ✧ UARTwith hardware flow control and 7816 protocol support.
   - ✧ USB.
   - ✧ Swire debug Interface.
7) Up to 6 channels of differential PWM:
   - ✧ PWM1~PWM5: 5-channel normal PWM output.
   - ✧ PWM0: 1channel with IR/IR FIFO mode for IR generation without MCU intervention.
8) Sensor:
   - ✧ 14bit SAR ADC with PGA
   - ✧ Temperature sensor
9) Onequadrature decoder.
10) Embedded low power comparator and continuous comparator.
11) Embedded hardware AES and AES-CCM.
12) Embedded hardware acceleration for Elliptical curve cryptography (ECC) used in HomeKit, Thread, and BLE4.2 and above.
13) Operating temperature:
   - ✧ ET versions: -40℃~+85℃
   - ✧ AT versions: -40℃~ +125℃

14) Supportsall 2.4GHz IoT standards into a single SoC, includingBLE, BLE Mesh, Zigbee, RF4CE, HomeKit, 6LowPAN, Thread, ANT, and 2.4GHz proprietary technologies without the requirement for an external DSP.

1.2.2 RF Features

RF features include:

1) BLE/802.15.4/2.4GHz RF transceiver embedded, working in worldwide 2.4GHz ISM band.
2) Bluetooth 5.0 Compliant,1Mbps,2Mbps LE Enhancement FIPD version, 125Kbps and 500Kbps.
3) IEEE802.15.4 compliant, 250Kbps.
4) 2.4GHz proprietary1Mbps/2Mbps mode with Adaptive Frequency Hopping feature support.
5) ANT mode.
6) Rx Sensitivity: -93dBm@BLE 1Mbps, -97dBm@ IEEE802.15.4 250Kbps, -88dBm @ BLE 2Mbps mode.
7) Tx output power:up to +8dBm (supply power by lithium battery directly)/+5dBm (supply power by DCDC)/+10dBm (Boost Mode).
8) Single-pin antenna interface.
9) RSSI monitoringwith +/-1dB resolution.
10) Auto acknowledgement, retransmission and flow control.
11) Support pre-standard BLE location features including AoA (Angel of Arrival) and AoD (Angle of Departure) in BLE 1M or 2M mode.

1.2.3 Features of power management module

Features of power management module include:

1) Embedded DCDC.
2) Battery monitor: Supports low battery detection.
3) Power supply: 1.8V~4.3V.
4) Multiple stage power management to minimize power consumption.
5) Low power consumption(@3V):
   - ✧ RX mode: 3.8mA, 4.25mA (Boost Mode)
   - ✧ TX mode: 4.05mA @ 0dBm, 6.2mA @ 5dBm
   - ✧ MCU current: 2.5mA @ 48MHz
   - ✧ Deep sleep with SRAM retention: 1.4uA
   - ✧ Deep sleep with external wakeup (without SRAM retention): 0.4uA
   - ✧ Shutdown: 60nA

1.2.4 USB features

USB features include:

1) Compatible with USB2.0 Full speed mode.
2) Supports 9 endpoints including control endpoint 0 and 8 configurable data endpoints.
3) Independent power domain.
4) Supports ISP (In-System Programming) via USB port.

1.2.5 Flash features

The ST17H58 embeds Flash with features below:

1) Total 512KB (4Mbits).
2) Flexible architecture: 4KB per Sector, 64KB/32KB per block.
3) Up to 256 Bytes per programmable page.
4) Write protect all or portions of memory.
5) Sector erase (4KB).
6) Block erase (32KB/64KB).
7) Cycle Endurance: 100,000 program/erases.
8) Data Retention: typical 20-year retention.
9) Multi firmware encryptionmethods for anti-cloning protection.

1.2.6 RF4CE features

RF4CE features include:

1) Based on IEEE 802.15.4 Standard, certified RF4CE platform, with ZRC1.1/ZRC2.0 and MSO profile support.
2) Various transmission options including broadcast.
3) Provides a secured key generation mechanism.
4) Supports a simple pairing mechanism for devices with full application confirmation.
5) Only authorized devices are able to communicate.
6) Various power saving modes are supported for all device classes.
7) Supports AES-128bit encryption and AES-CCM (Counter with the CBC-MAC) mode.
8) Extensible to vendor specific profiles.
9) Lenze extended profile with audio support for voice command based searches.
10) Over the air (OTA) firmware upgrade with hardware support.

1.2.7 Zigbeefeatures

Zigbee features include:

1) Based on IEEE 802.15.4 Standard, certified Zigbee Pro and Zigbee 3.0 platform, with ZHA/ZLL profile and Zigbee 3.0 device support.
2) Uses multi-hop mesh networking to eliminate single points of failure and expand the reach of networks.
3) Allow low power operation, even support the Green Power feature.
4) Supports networks of thousands of nodes, providing a networking for the smart home or the smart city.
5) Uses a variety of security mechanisms, such as AES-128 encryption, device and network keys and frame counters.
6) Include all application level functionality of ZigBee Smart Energy.
7) Support seamlessinteroperability with a wide variety of smart devices.
8) Over the air (OTA) firmware upgrade with hardware support.

1.2.8 6LowPAN and Thread features

6LoWPAN and Thread features include:

1) Supports 6LowPAN, IPv6 and DHCPv6.
2) Supports UDP and DTLS.
3) Supports Thread v1.1 and up withThreadsecurity and commission.
4) Supports networks of 250 nodes or greater.

1.2.9 BLE Mesh features

BLE Mesh features include:

1) Compatible with Bluetooth SIG Mesh specification 1.0, with additional features from Lenze enhanced design.
2) Support flexible mesh control, e.g. N-to-1 and N-to-M.
3) Supports switch control for over 200 nodes without delay.
4) Supports real time status update for over 200 nodes.
5) Secure and safe control and scalable identification within network.
6) 8/16 groups can be controlled at the same time.
7) 128/256 nodes within mesh network.
8) Configurable to more or fewer hops (e.g. 4 hops) within mesh network, single hop delay less than 15ms.
9) Flexible RF channel usage with both BLE advertising channels and data channels for good anti-interference performance.

1.2.10 Concurrent mode feature

In concurrent mode, the chip supports multiple standard working concurrently.

Typical combination is Bluetooth LE + 802.15.4 based standard (e.g. Zigbee, Thread, or 6LoWPAN): BLE and 802.15.4 based stacks can run concurrently with one application state based on time division technology, e.g. BLE stack and Thread stack will run alternately during the divided time slots.

1.2.11 HomeKit features

BLE Mesh features include:

1) Single-chip solution with hardware acceleration for all HomeKit security operations
2) Apple (pre-)certified Software Development Kit reference design
3) Conformant to latest HomeKitspecification (HAP v2.0)
4) Tested against Apple HomeKit Accessory Tester and Apple iOS 10 HomeKit HOME application
5) Support for all HAP defined services and characteristics
6) Support for custom defined HAP services and characteristics
7) HomeKit custom update over-the-air (OTA) profile for secure software upgrade over theair implemented.

1.3 Typical applications

The ST17H58 can be applied to IoT(Internet of Things) and HID (Human Interface Devices) applications, such as BLE smart devices, BLE mesh devices, 6LoWPAN/Thread home automation devices, 2.4GHz IEEE 802.15.4, RF4CE remote control /set-top box, and Zigbee systems.

Its typical applications include, but are not limited to the following:

- ✧ Smartphone and tablet accessories;
- ✧ RFRemote Control;
- ✧ Sports and fitness tracking;
- ✧ Wearable devices;
- ✧ Wireless toys;
- ✧ Smart Lighting, Smart Home devices;
- ✧ Building Automation;
- ✧ Smart Grid;
- ✧ Intelligent Logistics/Transportation/City;
- ✧ Consumer Electronics；
- ✧ Industrial Control;
- ✧ Health Care.

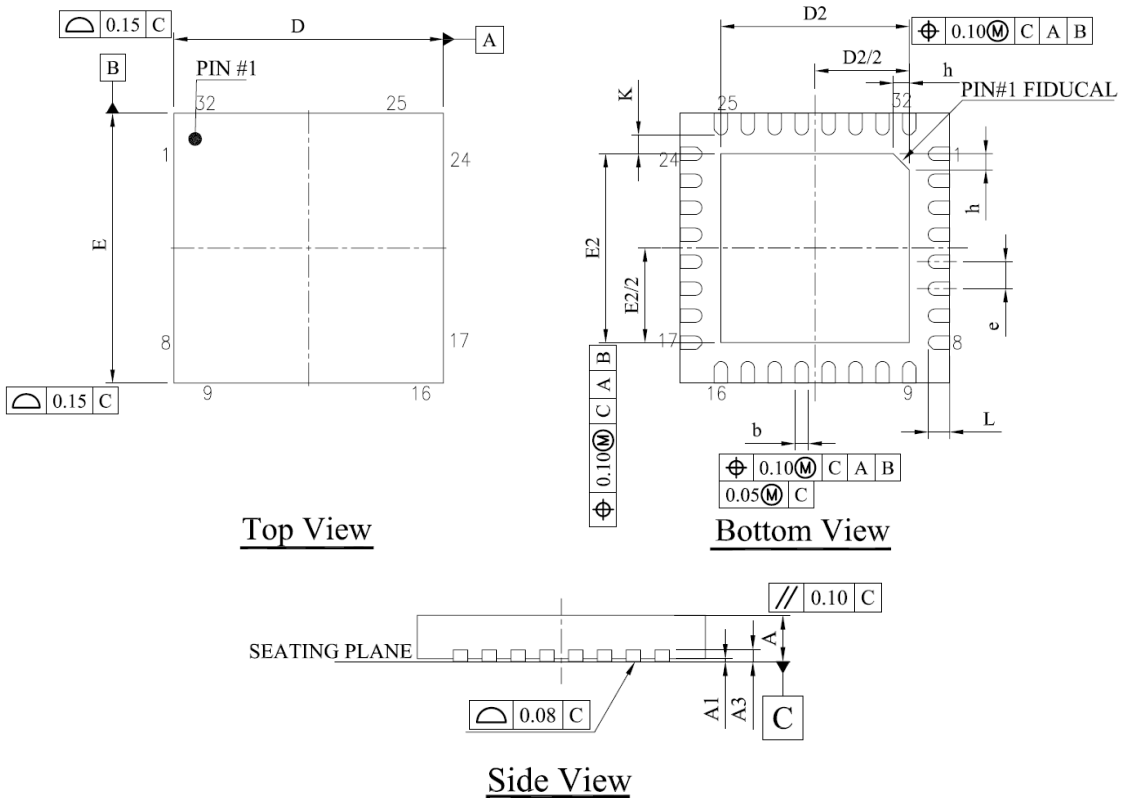
1.4 Ordering information

Table 1- 1 Ordering information of the ST17H58

Product Series	Package Type	Temperature Range	Product Part No.	Packing Method	Ordering Number	MinimumOrder Quantity
ST17H58	32-pin 5x5mm TQFN	-40℃~+85℃	ST17H58 ET32	TR	ST17H58 ET32R	3000
	32-pin 5x5mm TQFN	-40℃~+125℃	ST17H58 AT32	TR	ST17H58 AT32R	3000

1.5 Package

Package dimensionsfor the ST17H58,are shown as below.



SYMBOL	DIMENSION (MM)			DIMENSION (MIL)		
	MIN.	NOM.	MAX.	MIN.	NOM.	MAX.
A	0.70	0.75	0.80	27.6	29.5	31.5
A1	0	0.02	0.05	0	0.8	2.0
A3	---	0.20REF	---	---	7.9REF	---
b	0.18	0.25	0.30	7.1	9.8	11.8
D	4.90	5.00	5.10	192.9	196.9	200.8
D2	3.40	3.50	3.60	133.9	137.8	141.7
E	4.90	5.00	5.10	192.9	196.9	200.8
E2	3.40	3.50	3.60	133.9	137.8	141.7
e	---	0.50TYP	---	---	19.7TYP	---
K	0.20	---	---	7.9	---	---
L	0.35	0.40	0.45	13.8	15.7	17.7
h	0.30	0.35	0.40	11.8	13.8	15.7

NOTE:
1. DIMENSIONING AND TOLERANCING CONFORM TO ASME Y14.5M-1994.
2. POD REF BASED ON CUSTOMER SPECS.
3. DIMENSION "b" APPLIES TO METALLIZED TERMINAL AND IS.
   MEASURED BETWEEN 0.18 AND 0.30mm FROM TERMINAL TIP.
4. LEADFRAME MATERIAL IS 194FH AND THICKNESS IS 0.203MM (8 MIL).
5. DIMENSION"D"&"E" WILL INCLUDE ALL SIDE BURR INDUCED DURING ASSEMBLY.

Figure 1- 2 Package dimension for the ST17H58 ET/AT32(Unit: mm)

1.6 Pin layout

1.6.1 Pin layout for ST17H58

Figure 1- 2 shows pin assignment for the ST17H58.

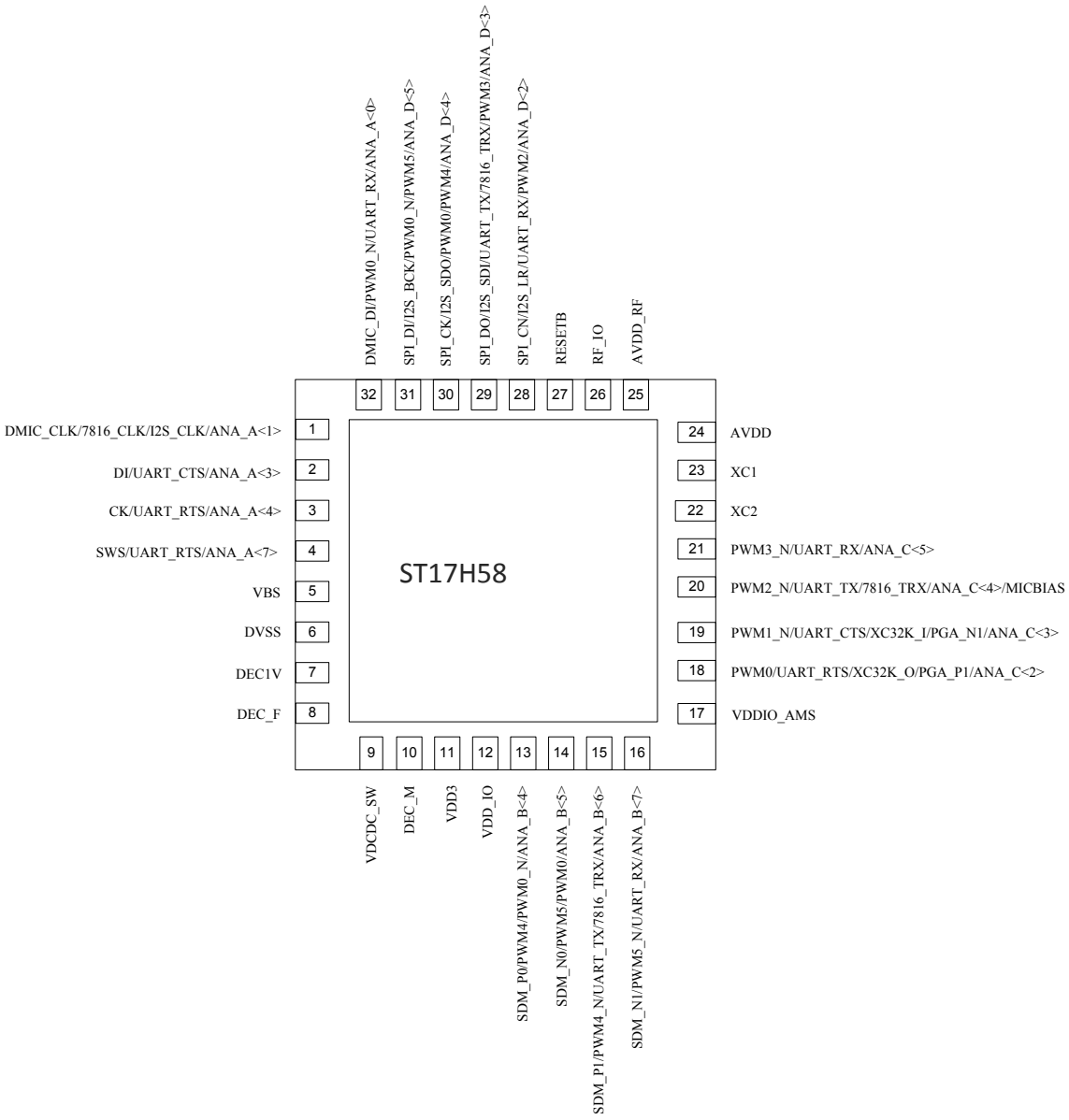


Figure 1- 2 Pin assignment for ST17H58

Functions of 32 pins for the ST17H58 are described inTable 1- :

Table 1- 2 Pin functions for ST17H58 ET32

No.	Pin Name	Type	Description
1	DMIC_CLK/7816_CLK /I2S_CLK/ANA_A<1>	Digital I/O	DMIC clock / UART 7816 clock /I2S clock / PA[1]

No.	Pin Name	Type	Description
2	DI/UART_CTS/ANA_A<3>	Digital I/O	SPI data input (I2C_SDA) /UART_CTS/ PA[3]
3	CK/UART_RTS/ANA_A<4>	Digital I/O	SPI clock (I2C_SCK) / UART_RTS / PA[4]
4	SWS/UART_RTS/ANA_A<7>	Digital I/O	Single wire slave/ UART_RTS / PA[7]
5	VBS	PWR	Battery voltage supplyfor chip
6	DVSS	GND	Digital LDO ground
7	DEC1V	PWR	VDDDEC (Digital LDO 1.2V output)
8	DEC_F	PWR	Connectexternal1uFcapacitor
9	VDCDC_SW	Analog	Connect with DEC_M via external 4.7uH inductor
10	DEC_M	Analog	Connect with VDCDC_SW via external 4.7uH inductor
11	VDD3	PWR	3.3V power supply
12	VDD_IO	PWR	Power supply for IO
13	SDM_P0/PWM4/PWM0_N /ANA_B<4>	Digital I/O	SDM positive output 0 / PWM4 output / PWM0 inverting output / PB[4]
14	SDM_N0/PWM5/PWM0 /ANA_B<5>	Digital I/O	SDM negative output 0 / PWM5 output / PWM0 output / PB[5]
15	SDM_P1/PWM4_N/UART_TX /7816_TRX/ANA_B<6>	Digital I/O	SDM positive output 1 / PWM4 inverting output / UART_TX / UART 7816 TRX / PB[6]
16	SDM_N1/PWM5_N /UART_RX/ANA_B<7>	Digital I/O	SDM negative output 1 / PWM5 inverting output / UART_RX / PB[7]
17	VDDIO_AMS	PWR	Power supply for mixed signal
18	PWM0/UART_RTS/ XC32K_O/PGA_P1/ANA_C<2>	Digital I/O	PWM0 output/ UART_RTS / (optional) 32KHz crystal output / PGA input / PC[2]
19	PWM1_N/UART_CTS/ XC32K_I/PGA_N1/ANA_C<3>	Digital I/O	PWM1 inverting output/ UART_CTS / (optional) 32KHz crystal input / PGA input/PC[3]
20	PWM2_N/UART_TX /7816_TRX/ANA_C<4>/MICBIAS	Digital I/O	PWM2 inverting output/ UART_TX / UART 7816 TRX/PC[4]/ AMIC BIAS
21	PWM3_N/UART_RX/ANA_C<5>	Digital I/O	PWM3 inverting output/ UART_RX / PC[5]
22	XC2	Analog O	24MHz crystal output
23	XC1	Analog I	24MHz crystal input
24	AVDD	PWR	Analog 1.4V supply
25	AVDD_RF	PWR	Analog 1.4V supply
26	RF_IO	Analog O	RF antenna
27	RESETB	RESET	Power on reset, active low
28	SPI_CN/I2S_LR/ UART_RX /PWM2/ANA_D<2>	Digital I/O	SPI chip select (Active low) / I2S left right channel select / UART_RX /PWM2 output/ PD[2]
29	SPI_DO/I2S_SDI/ UART_TX /7816_TRX /PWM3/ANA_D<3>	Digital I/O	SPI data output/ I2S serial data input / UART_TX / UART 7816 TRX / PWM3 output / PD[3]
30	SPI_CK/I2S_SDO/PWM0 /PWM4/ANA_D<4>	Digital I/O	SPI clock (I2C_SCK) / I2S serial data output / PWM0 output / PWM4 output / PD[4]
31	SPI_DI/I2S_BCK/PWM0_N /PWM5/ANA_D<5>	Digital I/O	SPI data input (I2C_SDA) / I2S bit clock / PWM0 inverting output / PWM5 output / PD[5]
32	DMIC_DI/PWM0_N/	Digital I/O	DMIC data input/PWM0 inverting output/ UART_RX /

No.	Pin Name	Type	Description
	UART_RX/ANA_A<0>		PA[0]

1.6.2 Notes

1) All digital IOs including ANA_A<0>~ ANA_D<7> can be used as GPIOs and have configurable pull-up/pull-down resistor.

2) I2C: ANA_C<0> and ANA_C<1> can be used as I2C. And I2C can also be multiplexed with SPI interface, i.e. I2C_SDA/I2C_SCK can be multiplexed with SPI_DI/SPI_CK respectively.

3) AMIC/Audio PGA input: ANA_C<0>~ANA_C<3>.

4) RX_CYC2LNA & TX_CYC2PA: control enabling external PA/LNA. Please refer to section **3.1 Block diagram**.

5) CMP_D & CMP_DF: Continuous comparator data output and inverting output. Please refer to section **15** for continuous comparator.

6) Pin drive strength: ANA_A<0:4>, ANA_C<6:7> and ANA_D<0:7> support drive strength up to 4mA (4mA when “DS”=1, 2mA when “DS”=0); ANA_A<5:7>, ANA_B<0:3> and ANA_C<0:5> support strength up to 8mA (8mA when “DS”=1, 4mA when “DS”=0); ANA_B<4:7> support strength up to 16mA (16mA when “DS”=1, 12mA when “DS”=0). “DS” configuration will take effect when the pin is used as output. Please refer to section **7.1.1 Basic configuration** for the corresponding “DS” register address and the default setting.

2 Memory and MCU

2.1 Memory

The ST17H58 embeds 48KBSRAM(including two independent 8KB SRAMs with retention in deep sleep, and one 32KB SRAM without retention in deep sleep) as data memory, and 512KB internal FLASH as program memory.

2.1.1 SRAM/Register

SRAM/Register memory map is shown as follows:

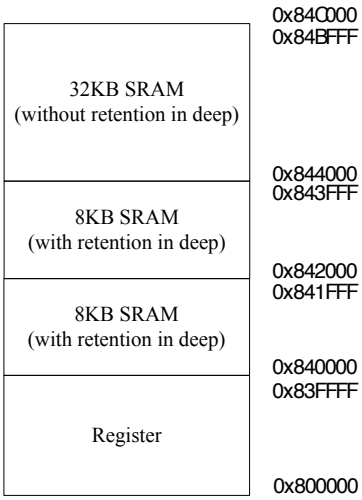


Figure 2- 1 Physical memory map

Register address: 0x800000 ~0x83FFFF.

Address for two independent 8KB SRAMs with retention in deep sleep: 0x840000 ~ 0x841FFF, 0x842000 ~ 0x843FFF.

Address for 32KB SRAM without retention in deep sleep: 0x844000 ~ 0x84BFFF.

Both register and SRAM address can be accessed (read or write)viadebugging interface (SWS/SWM, SPI/I2C/USB interface).

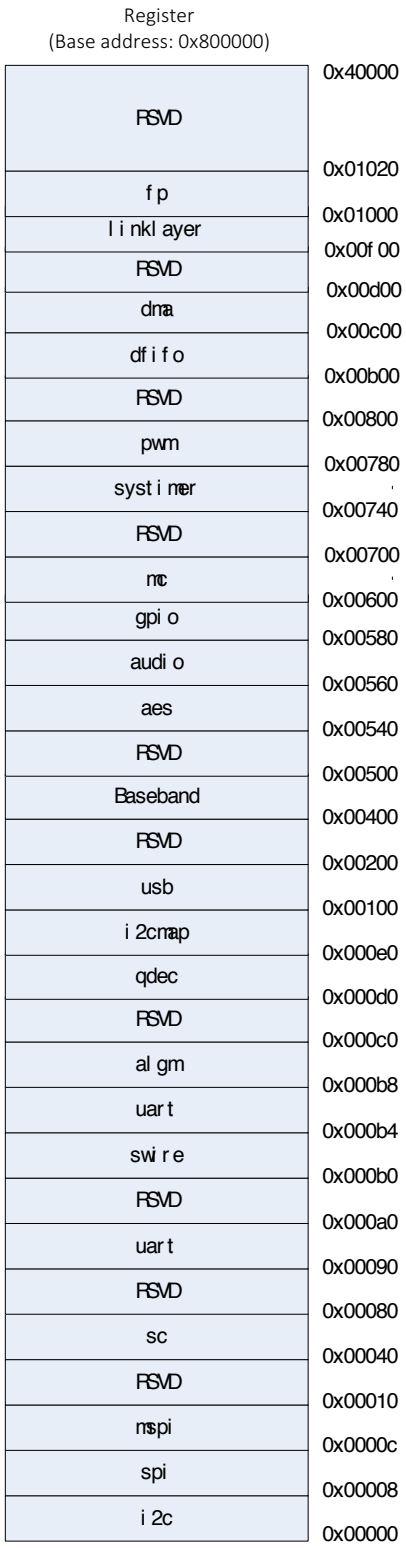


Figure 2- 2 Register space

2.1.2 Flash

The internal Flash mainly supports page program, sector/block/chip erase operations, and deep power down operation.Please refer to "***AN_RSPH-E1_Lenze RF SoC Programming Handbook***" and "***AN_15070101_Lenze Internal 512KB (4Mbits) Flash Operation Manual***" for Lenze Flash memory details.

For chip identification and traceability, the Flash is preloaded with 3-byte or 7-byte hexadecimal Unique Chip ID (UID) . User is not allowed to modify this preloaded UID, but can read the UID via corresponding API interface.

2.2 Firmware encryption

The ST17H58 supports multiple firmware encryption methods to achieve the anti-cloning protection, including:

- ✧ UID-based authentication code generation method

  During firmware burning (e.g. via specific burning jig), user can use customized key and AES encryption algorithm to encrypt the UID read from the chip flash, generate unique ciphertext and write the ciphertext into E-Fuse section.

  During application, an encryption authentication procedure is added. User should use the same key and AES encryption algorithm to encrypt the UID read from the chip flash, and generate new ciphertext. Before running main application firmware, the new ciphertext will be compared with the ciphertext read from the E-Fuse section. Only when the authentication passes, i.e. the comparison result matches, the main firmware will be up and running, otherwise the chip will stop running the main firmware.

- ✧ Bootloader-based firmware encryption/decryption

  The firmware can be encrypted using a customer-provided security key.The customer security key is written into a specific secure register, and becomesunreadable. Any attempt to read the key will only result in either all 1's or all 0's.

  The encrypted firmware can be generated based on the plaintext firmware and the customer security key. The customer can burn the security key into the obscured memory area and also the encrypted firmware into Flash.

  The firmware is readable by all, but appears as garbled binaries to 3rd party.

2.3 MCU

The ST17H58 integrates a powerful 32-bit MCU developed by Lenze. The digital core is based on 32-bit RISC, and the length of instructions is 16 bits; four hardware breakpoints are supported.

2.4 Working modes

The ST17H58 supports six working modes, including Active, Idle, Suspend, Deep sleep with SRAM retention, deep sleep without SRAM retention, and Shutdown.

- ✧ The Power Management (PM) module is always active in all working modes.
- ✧ For modules such as MCU, RF transceiver (Radio), and SRAM, the state depends on working mode, as shown below.

Table 2- 1 Working modes

Mode	Active	Idle	Suspend	Deep sleep with SRAM retention	Deep sleep without SRAM retention	Shutdown
MCU	active	stall	stall	off	off	off
32K Normal SRAM (without retention in deep sleep)	on	on	on	off	off	off
Radio	available	available	off	off	off	off
USB	available	available	off	off	off	off
Audio	available	available	off	off	off	off
Wakeup time to Active mode	——	0us	100us	Shorter than deep sleep without retention, almost same as Suspend	1ms	10ms
Two 8K retention SRAMs (with retention in deep sleep)	full	full	full	full	off	off
Wakeup on RTC (32K Timer wakeup)	——	——	available	available	available	off
Wakeup on pin (IO wakeup)	——	——	available	available	available	off
Wakeup on comparator (low power comparator wakeup)	——	——	available	available	available	off
Wakeup on interrupt	——	available	——	——	——	——
Wakeup on reset pin (RESETB)	——	available	available	available	available	on
Current	TBD	TBD	TBD	TBD	TBD	TBD

***Notes:**

1) “active”: MCU is at working state.
2) “stall”: In Idle and Suspend mode, MCU does not work, while its clock is still running.
3) “available” for Modules: It’s selectable to be at working state, or stall/be powered down if it does not need to work.
4) “available”/“on” for wakeup: Corresponding wakeup method is supported.
5) “off” for wakeup: Corresponding wakeup method is not supported.
6) “on”/“off”/“full” for SRAMs:
   - ✧ “on”: The 32K SRAM is powered on and works normally (can be accessed) in Active, Idle

and Suspend mode.

- ✧ “full”: Full speed. In Active, Idle and Suspend mode, the two 8K retention SRAMs are powered on and work normally (can be accessed); in Deep sleep with SRAM retention, the retention SRAMs are powered on, however, the contents of the retention SRAMs can be retained and cannot be accessed.
- ✧ “off”: The 32K SRAM is powered down in two Deep sleep modes and Shutdown mode. The retention SRAMs are powered down in Deep sleep without SRAM retention and Shutdown mode.

7) Current:

- ✧ In Deep sleep without SRAM retention, only the PM module is active, all digital and analog modules are powered down, thus the power consumption is largely decreased.
- ✧ In Deep sleep with SRAM retention, the PM module is active, all analog and digital modules except for the two retention SRAMs are powered down, thus the power consumption is a little higher than in Deep sleep without SRAM retention, but much lower than in Suspend.

Please refer to section **2.6.2 Working mode switch** for details.

2.5 Reset

The chip supports three types of reset methods, including POR (Power-On-Reset), watchdog reset and software reset.

1) POR: After power on, the whole chip will be reset, and all registers will be cleared to their default values.
2) Watchdog reset: A programmable watchdog is supported to monitor the system. If watchdog reset is triggered, registers except for a few retention analog registers will be cleared.
3) Software reset: It is also feasible to carry out software reset for the whole chip or some modules.
   - ✧ Setting address 0x6f[5] as 1b’1 is to reset the whole chip.Similar to watchdog reset, registers except for a few retention analog registers will be cleared by chip software reset.
   - ✧ Addresses 0x60~0x62 serve to reset individual modules: if some bit is set to logic “1”, the corresponding module is reset.

Table 2- 2 Register configuration for software reset

Address	Mnemonic	Type	Description	Reset Value
0x60	RST0	R/W	Reset control, 1 for reset, 0 for clear [0]: SPI [1]: I2C [2]: UART [3]: USB [4]: PWM [5]: QDEC	00

Address	Mnemonic	Type	Description	Reset Value
			[6]: RSVD [7]: SWS	
0x61	RST1	R/W	[0] ZB [1] SYS_TIMER [2] DMA [3] ALGM [4] AES [5] adc [6] alg [7] RSVD	df
0x62	RST2	R/W	[0] RSVD [1] audio [2] dfifo [3] RSVD [4] RISC [5] MCIC [6:7] RSVD	00
0x6f	PWDNEN	W	[0]: suspend enable [5]: rst all (act as watchdog reset) [6]: rsvd (mcu low power mode) [7]: stall mcu trig If bit[0] set 1, then system will go to suspend. Or only stall mcu	

2.6 Power Management

The multiple-stage Power Management (PM) module is flexible to control power state of the whole chip or individual functional blocks such as MCU, RF Transceiver, and peripherals.

2.6.1 Power-On-Reset (POR) and Brown-out detect

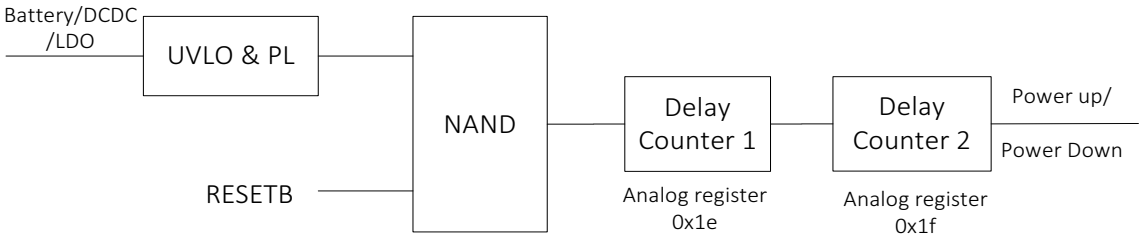


Figure 2- 3 Control logic for power up/down

The whole chip power up and down is controlled by the UVLO (Ultra-low Voltage Lockout) & PL (Power Logic) module and the external RESETB pin via the logic shown in the above diagram. UVLO takes the external power supply as input and releases the lock only when the power supply voltage is higher than a preset threshold. The RESETB pin has an internal pull-up resistor; an external Cap can be connected on the RESETB pin to control the POR delay.

After both UVLO and RESETB release, there are further two-level configurable delays (Delay

counter 1 and 2) before the system is released. The two delays are adjusted by analog register 0x1e and 0x1f, respectively. Since the contents of 0x1e~0x1f are reset to default only after power cycle, watchdog reset, or software reset, the delay change using 0x1e~0x1f is only applicable when the chip has not gone through these reset conditions. For example, after deep sleep wakeup, the settings in 0x1e~0x1f will take effect.

Table 2- 3 Analog register to control delay counters

Address	Name	Type	Description	Default
0x1e	r_dly0	R/W	initial power on delay, wait for VBUS LDO stable(64KHz count)	0x10
0x1f	r_dly1	R/W	wait for DCDC ready (64kHz count)	0x60

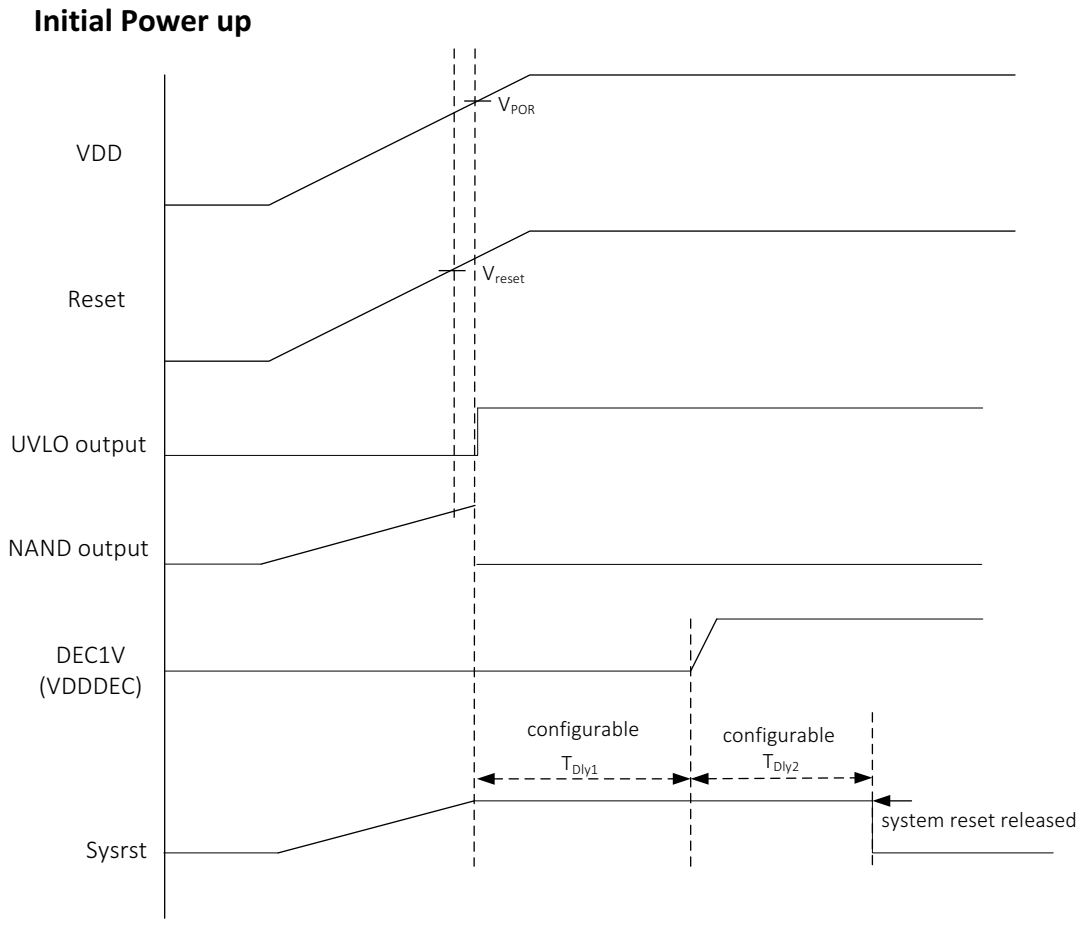


Figure 2-4 Initial Power-up sequence

Power down

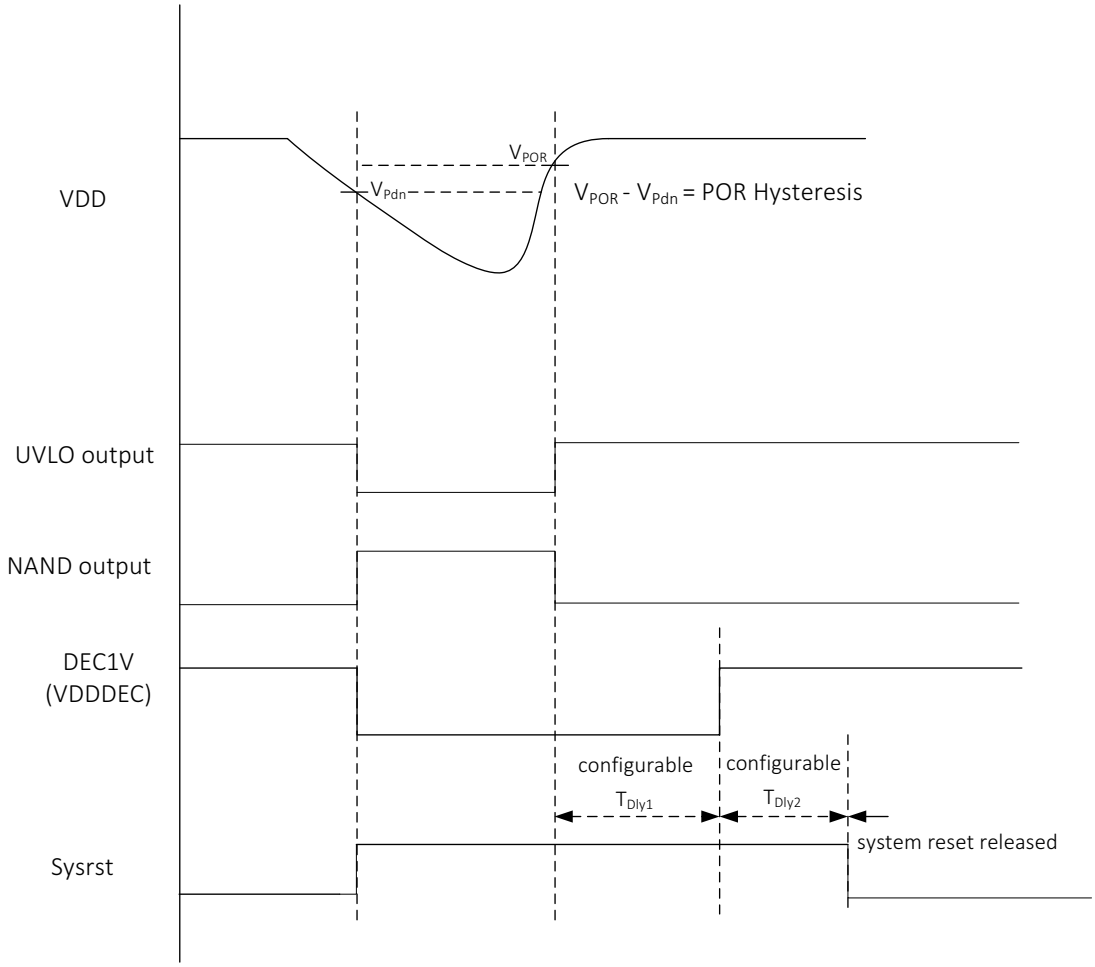


Figure 2- 5 Power-down sequence

Table 2- 4 Characteristics of Initial Power-up/ Power-down sequence

Symbol	Parameter	Min.	Typ.	Max.	Unit
V_{POR}	VDD voltage when V_{UVLO} turns to high level	TBD	TBD	TBD	V
V_{Pdn}	VDD voltage when V_{UVLO} turns to low level	TBD	TBD	TBD	V
T_{Dly1}	Delay counter1 value	Configurable via analog register 0x1e			
T_{Dly2}	Delay counter2 value	Configurable via analog register 0x1f			

2.6.2 Working mode switch

In Active mode, MCU is active, all SRAMs are accessible, and other modules are selectable whether to be at working state.

The chip can switch to Idle mode to stall the MCU. In this mode,all SRAMs are still accessible,

modules such as RF transceiver, Audio and USB are still selectable whether to be at working state. The chip can be triggered to Active mode by interrupt or RESETB pin, and the time to switch to Active mode is negligible.

To decrease power consumption to different levels, the chip can switch to power saving mode (Suspend, Deep sleep with SRAM retention, Deep sleep without SRAM retention, Shutdown) correspondingly. (Please refer to Table 2- 1.)

- ✧ In Suspend mode, MCU stalls, all SRAMs are still accessible, the PM module is active, modules such as RF transceiver, Audio and USB are powered down. The chip can be triggered to Active mode by 32K Timer, IO pin, low power comparator or RESETB pin. It takes 100us or so to switch from Suspend mode to Active mode.
- ✧ In Deep sleep with SRAM retention, the PM module is active, analog and digital modules except for the two 8K retention SRAMs are powered down, while the retention SRAMs can be retained and not accessible. The chip can be triggered to Active mode by 32K Timer, IO pin, low power comparator or RESETB pin. The time to switch to Active mode is shorter than Deep sleep without SRAM retention and close to Suspend.
- ✧ In Deep sleep without SRAM retention, only the PM module is active, while analog and digital modules including the retention SRAMs are powered down. The chip can be triggered to Active mode by 32K Timer, IO pin, low power comparator or RESETB pin. The time to switch to Active mode is 1ms or so.
- ✧ In Shutdown mode, all digital and analog modules are powered down, and only the PM module is active. The chip can be triggered to Active mode by RESETB pin only. The time to switch to Active mode is 10ms or so.

User can directly invoke corresponding library function to switch working mode of the chip.

If certain module doesn't need to work, user can power down this module in order to save power.

Table 2- 53.3V analog registers for module power up/down control

Address	Local name	Default	Description
afe3V_reg05<0>	32K_rc_pd	0	Power down 32KHz RC oscillator 1: Power down, 0: Power up
afe3V_reg05<1>	32k_xtal_pd	1	Power down 32k crystal 1: power down, 0: power up
afe3V_reg05<2>	24M_rc_pd	0	Power down of 24MHz RC oscillator 1: Power down, 0: Power up
afe3V_reg05<3>	xtal_LDO_pd	0	Power down of 16MHz crystal oscillator 1: Power down, 0: Power up
afe3V_reg05<4>	pd_pl_all_3v	0	Power down of power logic, 4.2V VBUS_LDO and DCDC 1: Power down, 0: Power up
afe3V_reg05<5>	pd_pl_dcdc_3v	0	Power down of DCDC 1: Power down, 0: Power up

Address	Local name	Default	Description
afe3V_reg05<6>	pd_pl_vbus_ldo_3v	0	Power down of VBUS_LDO 1: Power down, 0: Power up
afe3V_reg05<7>	ana_ldo_pd_3V	0	Power down baseband pll LDO 1: Power down, 0: Power up
afe3V_reg06<0>	rx_digLDO_pd	1	Power down rx dig LDO in RF transceiver 1: Power down, 0: Power up
afe3V_reg06<1>	rx_lnaLDO_pd	1	Power down LNA LDO in RF transceiver 1: Power down, 0: Power up
afe3V_reg06<2>	rx_anaLDO_pd	1	Power down analog LDO in RF transceiver 1: Power down, 0: Power up
afe3V_reg06<3>	rx_rfLDO_pd	1	Power down RF LDO in RF transceiver 1: Power down, 0: Power up
afe3V_reg06<4>	pll_BG_pd	1	Power down Bandgap in PLL 1: Power down, 0: Power up
afe3V_reg06<5>	pd_bypass_sw_3v	1	Power down of bypass switch in 4.3V power domain 1: Power down, 0: Power up
afe3V_reg06<6>	pll_vco_ldo_pd	1	Power down VCO LDO 1: Power down, 0: Power up
afe3V_reg06<7>	pll_cp_ldo_pd	1	Power down cp and prescaleranaog circuit ldo 1: Power down, 0: power up
afe3V_reg07<0>	spd_ldo_pd	1	Power down of spdldo 1: Power down, 0: power up
afe3V_reg07<1>	dig_ldo_pd	0	Power down of main digital ldo 1: Power down, 0: power up
afe3V_reg07<2>	dig_ret_pd	1	Power down of retention ldo 1: Power down, 0: power up
afe3V_reg07<3>	pd_lc_comp_3v	1	Power down of low power comparator: 1: Power down, 0: Power up
afe3V_reg07<4>	pd_temp_sensor_3v	1	Power down of temp sensor: 1: Power down, 0: Power up

2.6.3 LDO and DCDC

The chip embedded DCDC can generate 2.7~3.6V output voltage and supply power for the internal flash; the DCDC can also generate 1.4V output voltage.

The embedded LDO regulator takes the 1.4V voltage output from the DCDC, and generates 1.2V regulated voltage to supply power for 1.2V digital core and analog modules in Active/Idle/Suspend mode.

2.7 Wakeup sources

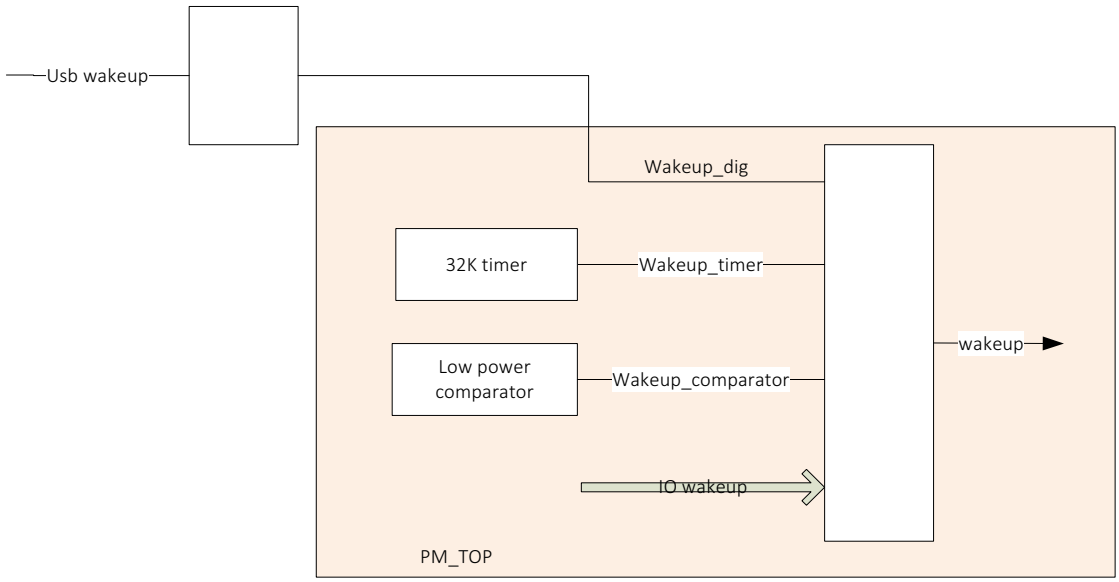


Figure 2- 6 Wakeup sources

2.7.1 Wakeup source - USB

This wakeup source can only wake up the system from suspend mode.

First, set the digital register 0x6e bit[2] as 1b'1.

To activate this mode, analog register0x26[5] should also be set as 1b'1.

Once USB host sends out resuming signal, the system will be woke up.

2.7.2 Wakeup source – 32K timer

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

To enable the wakeup source from 32K timer, analog register 0x26[6] should be set as 1b'1.

2.7.3 Wakeup source – low power comparator

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes.

To enable the wakeup source from low power comparator, analog register0x26[7] should be set as 1b'1. The low power comparator wakeup is active high.

2.7.4 Wakeup source – IO

This wakeup source is able to wake up the system from suspend mode or two deep sleep modes. And IO wakeup supports high level or low level wakeup which is configurable via polarity control registers.

Analog register 0x26[4] should be set as 1b'1 to enable IO wakeup source.

Enabling control analog registers: PA[7:0] enabling control register is 0x27[7:0], PB[7:0] enabling control register is 0x28[7:0], PC[7:0] enabling control register is 0x29[7:0], and PD[7:0] enabling control register is 0x2a[7:0]. Total wakeup pin can beup to 32.

Polarity control registers: PA[7:0] polarity control register is 0x21[7:0],PB[7:0] polarity control register is 0x22[7:0], PC[7:0] polarity control register is 0x23[7:0], and PD[7:0] polarity control register is 0x24[7:0].

The corresponding driver is available so that user can directly invoke it to use IO wakeup source.

Analog register 0x44[3:0] indicates the wakeup source which triggers system wakeup. After wakeup, the corresponding wakeup status will be set as 1b'1 automatically, and it's needed to write 1 to manually clean the status.

2.7.5 Register table

Table 2- 6 Analog registers for Wakeup

Address	Name	Type	Description	Default
0x21	PA_POL	R/W	Polarity control registers for IO wakeup	0x00
0x22	PB_POL	R/W		0x00
0x23	PC_POL	R/W		0x00
0x24	PD_POL	R/W		0x00
0x25	rsvd	R		
0x26	wkup_en	R/W	[7] low power comparator wake up enable	0x00
			[6] 32k timer wake up enable	
			[5] digital core (USB) wakeup enable	
			[4] IO (pad) wake up enable	
			[3:2] SRAM sleep enable. [2] low 8k SRAM. [3] high 8k SRAM.	
			[1:0] SRAM retention enable. [0] low 8k SRAM. [1] high 8k SRAM.	
0x27	PA wake up enable	R/W	Enabling control registers for IO wakeup	0x00
0x28	PB wake up enable	R/W		0x00
0x29	PC wake up enable	R/W		0x00
0x2a	PD wake up enable	R/W		0x00
0x44	status	R	[7] xtal_rdy	
			[6] wd_status	
			[5]cal_done_24m	
			[4]cal_done_32k	
			[3] IO (pad) wakeup status	
			[2]digital core (USB) wakeup status	
			[1] 32k timer wakeup status	
			[0] low power comparator wakeup status	

Table 2- 7 Digital register for Wakeup

Address	Mnemonic	Type	Description	Reset Value
0x6e	WAKEUPEN	R/W	Wakeup enable [0]: enable wakeup from I2C host [1]: enable wakeup from SPI host [2]: enable wakeup from USB [3]: enable wakeup from gpio [4]: enable wakeup from I2C synchronous interface System resume control [5]: enable GPIO remote wakeup [6]: if set to1, system will issue USB resume signal on USB bus [7] sleep wakeup reset system enable	00

3 BLE/802.15.4/2.4G RF Transceiver

3.1 Block diagram

The ST17H58 integrates an advanced BLE/802.15.4/2.4GHz RF transceiver. The RF transceiver works in the worldwide 2.4GHz ISM (Industrial Scientific Medical) band and contains an integrated Balun with a single-ended RF Tx/Rx port pin. No matching components are needed.

The transceiver consists of a fully integrated frequency synthesizer, a power amplifier, a modulator and a receiver. The transceiver can be configured to work in standard-compliant 1Mbps BLE mode,2Mbps enhancement BLE mode, 125Kbps BLE mode, 500Kbps BLE mode, IEEE802.15.4 standard-compliant 250Kbpsmode, and Proprietary 1Mbps and 2Mbps mode. All modes support FSK/GFSK/MSK modulations.

To control external PA and LNA, first follow the GPIO lookup table (see section 7.1.1.3 GPIO lookup table) to configure the specific two pins as TX_CYC2PA and RX_CYC2LNA function, respectively (Note that other functions with higher polarity should be disabled at the same time) .After the two pins are configured as TX_CYC2PA and RX_CYC2LNA function, the output function is enabled. The two pins are high active: When both the two pins output low level, the external PA and LNA are disabled; when one of the two pins output high level, the external PA/LNA are enabled correspondingly; the two pins won't output high level simultaneously.

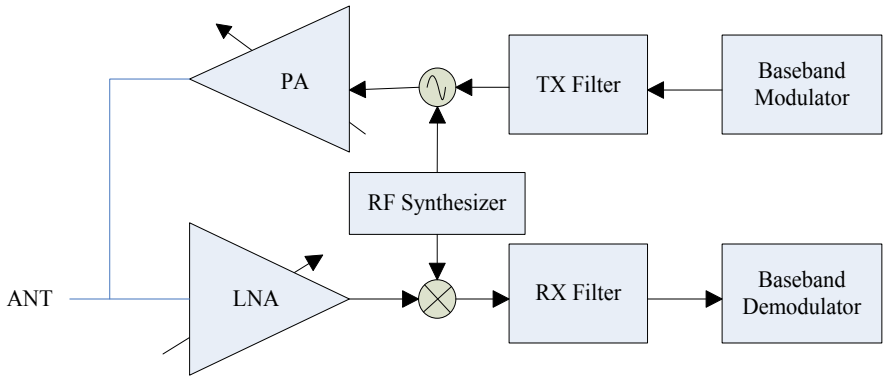


Figure 3- 1 Block diagram of RF transceiver

The internal PA can deliver a maximum 8dBm output power, avoiding the needs for an external RF PA.

3.2 Function description

3.2.1 Turn on/off

For the sake of saving power, the transceiver can be turned on/off via the software. Setting the address 0x7c bit[6] to 1 enables the RF transceiver, while clearing the bit totally disables the RF transceiver.

3.2.2 Air interface data rate and RF channel frequency

Air interface data rate, the modulated signaling rate for RF transceiver when transmitting and receiving data, is configurable via related register setting: 125Kbps, 250Kbps, 500Kbps, 1Mbps, 2Mbps.

For the ST17H58, RF transceiver can operate with frequency ranging from 2.400GHz to

2.4835GHz. The RF channel frequency setting determines the center of the channel.

3.3 Baseband

The baseband is disabled by default. The corresponding API is available for user to power on/down the baseband and enable/disable clock, so that the baseband can be turned on/off flexibly.

The baseband contains dedicated hardware logic to perform fast AGC control, access code correlation, CRC checking, data whitening, encryption/decryption and frequency hopping logic.

The baseband supports all features required by Bluetooth V5.0and 802.15.4 specification.

3.3.1 Packet format

Packet format in standard 1Mbps BLE mode is shown as Table 3- 1:

Table 3- 1 Packet Format in standard 1Mbps BLE mode

LSB MSB

Preamble (1 octet)	Access Address (4 octets)	PDU (2 ~ 257 octets)	CRC (3 octets)

Packet length 80bit ~ 2120bit (80~2120us @ 1Mbps).

Packet format in standard 2Mbps BLE mode is shown as Table 3- 2:

Table 3- 2 Packet format in standard 2Mbps BLE mode

LSB MSB

Preamble (2 octets)	Access Address (4 octets)	PDU (2 ~ 257 octets)	CRC (3 octets)

Packet format in standard 500Kbps/125Kbps BLE mode is shown as Table 3- 3:

Table 3- 3 Packet format in standard 500Kbps/125Kbps BLE mode

LSB MSB

Preamble (10 octets)	Access Address (4 octets)	CI (2 bits)	TERM1 (3 bits)	PDU (2 ~ 257 octets)	CRC (3 octets)	TERM2 (3 bits)

Packet format in 250Kbps 802.15.4 mode is shown asTable 3- 4:

Table 3- 4 Packet format in 802.15.4 mode

LSB MSB

Preamble (4~16 octets)	SFD (1 octet)	Frame length (1 octet)	PSDU (Variable0~127octets)	CRC (2octets)
SHR		PHR	PHY payload	

Packet format in 2.4Ghz Proprietarymode is shown asTable 3- 5:

Table 3- 5 Packet format in Proprietary mode

LSB MSB

Preamble (8 bits)	Address code (configurable 3~5 bytes)	Packet Controller + Payload (1~33 bytes)	CRC (1~2 bytes)

3.3.2 BLE location function

In BLE 1M or 2M mode, BLE location features including AoA (Angel of Arrival) and AoD(Angle of Departure) are supported.

In the location mode of operation, the chip transmits a training sequence concatenated to the normal packet transmissions. In AoA mode of operation, the receiving side has multiple antennas and will be switched during the training sequence period. In AoD mode of operation, the transmitting side has multiple antennas and will be switched during the training sequence period. In either mode, the receiving side will be able to determine based on the phase variations of the received training sequences, the angle of location of the peer device.

3.3.3 RSSI and frequency offset

The ST57H25F512provides accurate RSSI (Receiver Signal Strength Indicator) and frequency offset indication.

- ✧ RSSI can be readfromthe 1byte at the tail of each receiveddatapacket.
- ✧ If no data packet is received (e.g. carrier or interferingWi-Fi signal is detected), real-time RSSI can also be read from specific registerswhich will be updated automatically.
- ✧ Frequency offset can be read from the 2bytes at the tail of the data packet. Valid bits of actual frequency offset may be less than 16bits, and different valid bits correspond to different tolerance range.

Lenze supplies corresponding drivers for user to read RSSI and frequency offset as needed.

4 Clock

4.1 Clock sources

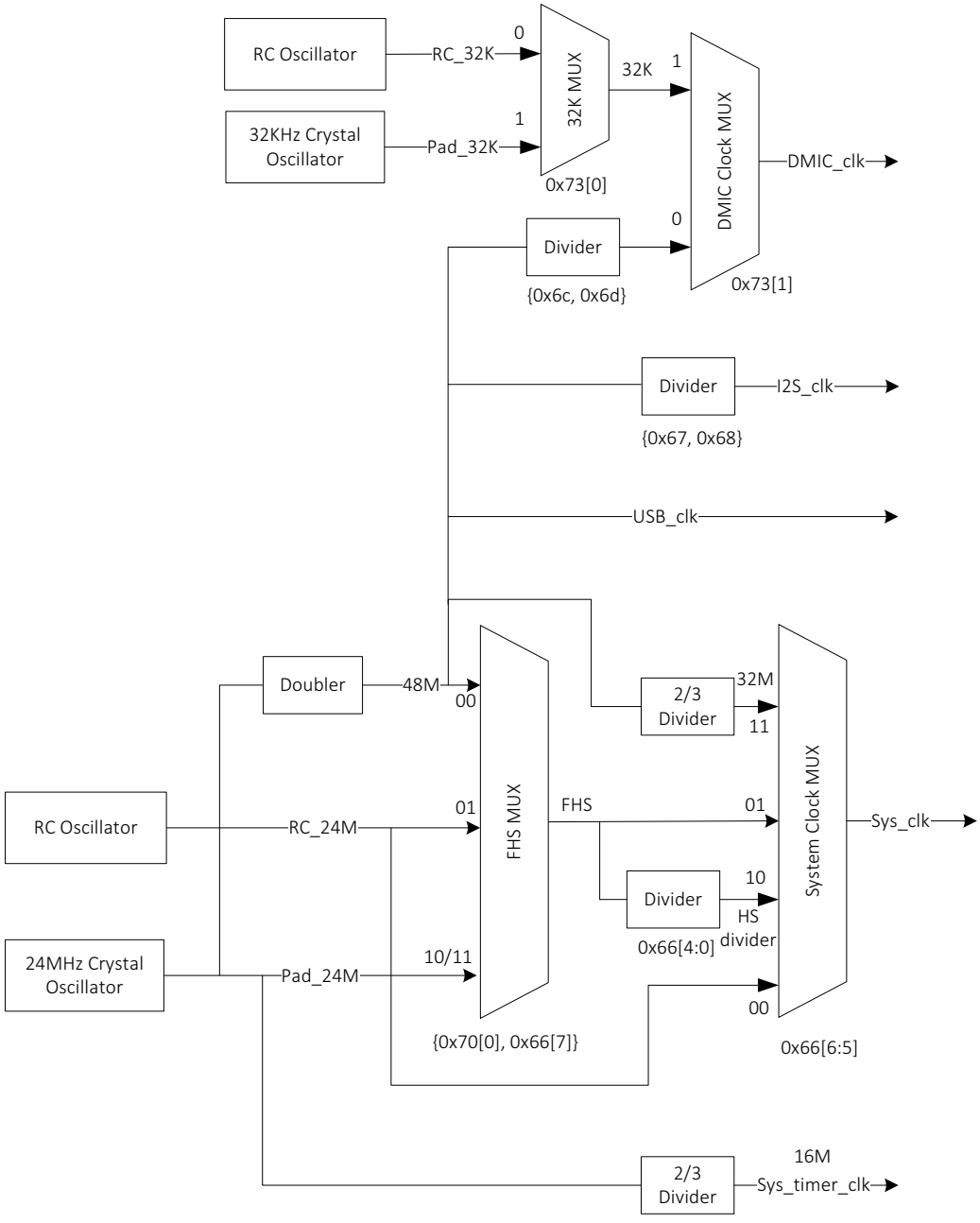


Figure 4- 1 Block diagram of clock

The ST17H58 embeds a 24MHz RC oscillator which can be used as clock source for system, as well as a 32KHz RC oscillator to provide clock source for DMIC and sleep state.

External 24MHz crystal is available via pin XC1, which can provide a Pad_24MHz clock source for system and System Timer, and generate a 48M clock via a frequency doubler to provide clock source for DMIC, I2S and USB. External 32K crystal is available via pin ANA_C<3>, which can provide a 32KHz clock source for DMIC.

4.2 System clock

There are four selectable clock sources for MCU system clock: **RC_24M** derived from 24MHz RC oscillator, High speed clock "**FHS**", **HS divider clock** (derived from "FHS" via a frequency divider), and **32MHz clock** derived from 48MHz clock via a 2/3 frequency divider (The 48M clock is derived from 24M crystal oscillator via a frequency doubler).

The high speed clock (FHS) is selectable via address {0x70[0], 0x66[7]}from the following sources: **48MHz** clock (derived from 24M crystal oscillator via a frequency doubler), **RC_24M** (derived from 24MHz RC oscillator), and **Pad_24M** (derived from 24M crystal oscillator).

The digital register CLKSEL (address 0x66) serves to set system clock: System clock source is selectable via bit[6:5].

✧ If address 0x66[6:5] is set to 2b'10 to select the HS divider clock, system clock frequency is adjustable via address 0x66[4:0]. The formula is shown as below:

$F_{System\ clock} = F_{FHS}$ / (system clock divider value in address 0x66[4:0] + 1).

4.3 Module clock

Registers CLKEN0~CLKEN2 (address 0x63~0x65) are used to enable or disable clock for various modules. By disable the clocks of unused modules, current consumption could be reduced.

4.3.1 System Timer clock

System Timer clock is derived from 24M crystal oscillator via a 2/3 frequency divider. The clock frequency is fixed as 16MHz.

4.3.2 USB clock

USB clock is derived from 48M clock. The 48M clock is derived from 24M crystal oscillator via a frequency doubler.

4.3.3 I2S clock

I2S clock is derived from 48M clock via a frequency divider. The 48M clock is derived from 24M crystal oscillator via a frequency doubler.

Address 0x67[7] should be set as 1b'1 to enable I2S clock. I2S clock frequency dividing factor contains step and mod. Address 0x67[6:0] and 0x68 serve to set I2S clock step[6:0] and mod[7:0] respectively, and mod should be no less than 2*step.

I2S clock frequency, $F_{I2S\ clock}$, equals to 48M * I2S_step[6:0] / I2S_mod[7:0].

4.3.4 DMIC clock

Address 0x6c[7] serves to enable DMIC clock.

DMIC clock pin can select 32KHz clock, or derive from 48M clock via a frequency divider. Address 0x73 serves to select DMIC clock source.

✧ In normal DMIC working mode, 0x73[1] needs to be set as 1b'0, DMIC clock divider is selected and frequency dividing factor should be further configured. DMIC clock frequency dividing factor contains step and mod. Address 0x6c[6:0] and 0x6d serve to set DMIC clock step[6:0] and mod[7:0], respectively, and mod should be no less than 2*step.

In this situation, DMIC clock frequency, $F_{DMIC\ clock}$, equals to 48M * DMIC_step[6:0] / DMIC_mod[7:0].

✧ When DMIC is not used, and a 32KHz clock is needed, address 0x73 bit[1] should be set as 1b'1 to select the 32KHz clock, while bit[0] can be configured to select either RC_32K from 32K RC oscillator or Pad_32K from 32K crystal oscillator.

4.4 Register table

Table 4- 1 Register table related to clock

Address	Mnemonic	R/W	Description	Default
0x63	CLKEN0	R/W	Clock enable control: 1 for enable; 0 for disable [0]: SPI [1]: I2C [2]: UART [3]: USB [4]: PWM [5]: QDEC [6]: RSVD [7]: SWS	8c
0x64	CLKEN1	R/W	[0]ZB [1]SYS_TIMER [2]DMA [3]ALGM [4]AES [5:7]RSVD	00
0x65	CLKEN2	R/W	[0]RSVD [1]AUDIO [2]DFIFO [3]RSVD [4]MC [5]MCIC [6:7]RSVD	00
0x66	CLKSEL	R/W	System clock select [4:0]: system clock divider. If 0x66[6:5] is set as 2b'10, $F_{Sysclk} = F_{FHS} / (CLKSEL[4:0]+1)$. FHS: refer to 0x70 FHS_sel. [6:5]: select system clock source 2'b00: RC_24M from RC oscillator 2'b01: FHS 2'b10: HS divider (see 0x66[4:0]) 2'b11: 32M clock (48M * 2/3 divider) [7] FHS select (see 0x70[0])	ff

Address	Mnemonic	R/W	Description	Default
0x67	I2S step	R/W	[7] I2S clock enable [6:0] I2S step	33
0x68	I2S Mod	R/W	I2S mod I2S clock = 48M*I2S_step[6:0]/I2S_mod[7:0], Mod should be larger than or equal to 2*step.	2
0x6c	DMIC_step	R/W	[7]: Digital MIC clock enable [6:0] DMIC step	1
0x6d	DMIC_mod	R/W	[7:0] DMIC mod DMIC clock = 48M*DMIC_step[6:0]/DMIC_mod Mod should be larger than or equal to 2*step.	2
0x70	FHS_sel	R/W	{0x70[0], 0x66[7]} FHS select 2'b00: 48M clock doubled from 24M crystal 2'b01: RC_24M from RC oscillator 2'b1x: Pad_24M from 24M crystal oscillator	00
0x73	Clk mux sel		[0]: clk32k select 0: select RC_32k from RC oscillator 1: select Pad_32k from 32K crystal oscillator [1]: DMIC clock select 1: select 32k (see 0x73[0] to select 32K source) 0: select DMIC clk div (see 0x6c~0x6d) [2]: rsvd [7:4]: r_lpr_div, decide system clock speed in low power mode	

5 Timers

5.1 Timer0~Timer2

The ST57H25F512supports three timers: Timer0~ Timer2.The three timers all support four modes: Mode 0 (System Clock Mode), Mode 1 (GPIO Trigger Mode), Mode 2 (GPIO Pulse Width Mode) and Mode 3 (Tick Mode), which are selectable via the register TMR_CTRL0 (address 0x620) ~ TMR_CTRL1 (address 0x621).

Timer 2 can also be configured as “watchdog” to monitor firmware running.

5.1.1 Register table

Table 5- 1 Register configuration for Timer0~Timer2

Address	Mnemonic	Type	Description	Reset Value
0x72	Wd_status	R/W	[0] watch dog status,write 1 to clear.	
0x620	TMR_CTRL0	RW	[0]Timer0 enable [2:1] Timer0 mode. 0: using sclk, 1: using gpio, 2: count width of gpi, 3: tick [3]Timer1 enable [5:4] Timer1 mode. [6]Timer2 enable [7]Bit of timer2 mode	00
0x621	TMR_CTRL1	RW	[0]Bit of timer2 mode [7:1]Low bits of watch dog capture	00
0x622	TMR_CTRL2	RW	[6:0]High bits of watch dog capture. It is compared with [31:18] of timer2 ticker [7]watch dog capture	00
0x623	TMR_STATUS	RW	[0] timer0 status, write 1 to clear [1] timer1 status, write 1 to clear [2] timer2 status, write 1 to clear	
0x624	TMR_CAPT0_0	RW	Byte 0 of timer0 capture	00
0x625	TMR_CAPT0_1	RW	Byte 1 of timer0 capture	00
0x626	TMR_CAPT0_2	RW	Byte 2 of timer0 capture	00
0x627	TMR_CAPT0_3	RW	Byte 3 of timer0 capture	00
0x628	TMR_CAPT1_0	RW	Byte 0 of timer1 capture	00
0x629	TMR_CAPT1_1	RW	Byte 1 of timer1 capture	00
0x62a	TMR_CAPT1_2	RW	Byte 2 of timer1 capture	00
0x62b	TMR_CAPT1_3	RW	Byte 3 of timer1 capture	00
0x62c	TMR_CAPT2_0	RW	Byte 0 of timer2 capture	00
0x62d	TMR_CAPT2_1	RW	Byte 1 of timer2 capture	00
0x62e	TMR_CAPT2_2	RW	Byte 2 of timer2 capture	00
0x62f	TMR_CAPT2_3	RW	Byte 3 of timer2 capture	00
0x630	TMR_TICK0_0	RW	Byte 0 of timer0 ticker	
0x631	TMR_TICK0_1	RW	Byte 1 of timer0 ticker	
0x632	TMR_TICK0_2	RW	Byte 2 of timer0 ticker	
0x633	TMR_TICK0_3	RW	Byte 3 of timer0 ticker	

Address	Mnemonic	Type	Description	Reset Value
0x634	TMR_TICK1_0	RW	Byte 0 of timer1 ticker	
0x635	TMR_TICK1_1	RW	Byte 1 of timer1 ticker	
0x636	TMR_TICK1_2	RW	Byte 2 of timer1 ticker	
0x637	TMR_TICK1_3	RW	Byte 3 of timer1 ticker	
0x638	TMR_TICK2_0	RW	Byte 0 of timer2 ticker	
0x639	TMR_TICK2_1	RW	Byte 1 of timer2 ticker	
0x63a	TMR_TICK2_2	RW	Byte 2 of timer2 ticker	
0x63b	TMR_TICK2_3	RW	Byte 3 of timer2 ticker	

5.1.2 Mode0 (System Clock Mode)

In Mode 0, system clock is employed as clock source.

After Timer is enabled, Timer Tick (i.e. counting value) is increased by 1 on each positive edge of system clock from preset initial Tick value. Generally the initial Tick value is set to 0.

Once current Timer Tick value matches the preset Timer Capture (i.e. timing value), an interrupt is generated, Timer stops counting and Timer status is updated.

Steps of setting Timer0 for Mode 0 is taken as an example.

1st: Set initial Tick value of Timer0

Set Initial value of Tick viaregisters TMR_TICK0_0~TMR_TICK0_3 (address 0x630~0x633). Address 0x630 is lowest byte and 0x633 is highest byte. It's recommended to clear initial Timer Tick value to 0.

2nd: Set Capture value of Timer0

Set registers TMR_CAPT0_0~TMR_CAPT0_3 (address 0x624~0x627). Address 0x624 is lowest byte and 0x627 is highest byte.

3rd:Set Timer0 to Mode 0 and enable Timer0

Set register TMR_CTRL0 (address 0x620) [2:1] to 2b'00 to select Mode 0; Meanwhile set address 0x620[0] to 1b'1 to enable Timer0.Timer0 starts counting upward, and Tick value is increased by 1 on each positive edge of system clock until it reaches Timer0 Capture value.

5.1.3 Mode1 (GPIO Trigger Mode)

In Mode 1, GPIO is employed as clock source. The "**m0**"/"**m1**"/"**m2**" register specifies the GPIO which generates counting signal for Timer0/Timer1/Timer2.

After Timer is enabled, Timer Tick (i.e. counting value) is increased by 1 oneach positive/negative(configurable)edge of GPIO from preset initial Tick value. Generally the initial Tick value is set to 0. The "**Polarity**" register specifies the GPIO edge when Timer Tick counting increases.

Note: Refer to **Section 7.1.2** for corresponding "**m0**", "**m1**", "**m2**" and "**Polarity**" register address.

Once current Timer Tick value matches the preset Timer Capture (i.e. timing value), an interrupt is generated and timer stops counting.

Steps of setting Timer1 for Mode 1 is taken as an example.

1st: Set initial Tick value of Timer1

Set Initial value of Tick via registers TMR_TICK1_0~TMR_TICK1_3 (address 0x634~0x637). Address 0x634 is lowest byte and 0x637 is highest byte.It's recommended to clear initial Timer Tick value to 0.

2nd: Set Capture value of Timer1

Set registers TMR_CAPT1_0~TMR_CAPT1_3 (address 0x628~0x62b). Address 0x628 is lowest byte and 0x62b is highest byte.

3rd: Select GPIO source and edge for Timer1

Select certain GPIO to be the clock source via setting "m1" register.

Select positive edge or negative edge of GPIO input to trigger Timer1 Tick increment via setting "Polarity" register.

4th: Set Timer1 to Mode 1 and enable Timer1

Set address 0x620[5:4] to 2b'01 to select Mode 1; Meanwhileset address 0x620[3] to 1b'1 to enable Timer1. Timer1starts counting upward, and Timer1 Tick value is increased by 1 on each positive/negative (specified during the 3rd step) edge of GPIO until it reaches Timer1 Capture value.

5.1.4 Mode2 (GPIO Pulse Width Mode)

In Mode 2, system clock is employed as the unit to measure the width of GPIO pulse. The "**m0**"/"**m1**"/"**m2**" register specifies the GPIO which generates control signal for Timer0/Timer1/Timer2.

After Timer is enabled, Timer Tick is triggered by a positive/negative (configurable)edge of GPIO pulse. Then Timer Tick (i.e. counting value) is increased by 1 on each positive edge of system clock from preset initial Tick value. Generally the initial Tick value is set to 0. The "**Polarity**" register specifies the GPIO edge when Timer Tick starts counting.

Note: Refer to **Section 7.1.2** for corresponding "**m0**", "**m1**", "**m2**" and "**Polarity**" register address.

While a negative/positiveedge of GPIO pulse is detected, an interrupt is generated and timer stops counting. The GPIO pulse width could be calculated in terms of tick count and period of system clock.

Steps of setting Timer2for Mode 2 is taken as an example.

1st: Set initial Timer2 Tick value

Set Initial value of Tick via registers TMR_TICK2_0~TMR_TICK2_3 (address 0x638~0x63b). Address 0x638 is lowest byte and 0x63b is highest byte.It's recommended to clear initial Timer Tick value to 0.

2nd: Select GPIO source and edge for Timer2

Select certain GPIO to be the clock source via setting "m2" register.

Select positive edge or negative edge of GPIO input to trigger Timer2 counting start via setting "Polarity" register.

3rd: Set Timer2 to Mode 2 and enable Timer2

Setaddress 0x620[7:6] to 2b'01 and address 0x621 [0] to 1b'1.

Timer2 Tick is triggered by a positive/negative (specified during the 2nd step) edge of GPIO pulse. Timer2 starts counting upward and Timer2 Tick value is increased by 1 on each positive edge of system clock.

While a negative/positive edge of GPIO pulse is detected, an interrupt is generated and Timer2 tick stops.

4th: Read current Timer2 Tick value to calculate GPIO pulse width

Read current Timer2 Tick value from address 0x638~0x63b.

Then GPIO pulse width is calculated as follows:

$$\text{GPIO pulse width} = \text{System clock period} * (\text{current Timer2 Tick} - \text{intial Timer2 Tick})$$

For initial Timer2 Tick value is set to the recommended value of 0, then:

$$\text{GPIO pulse width} = \text{System clock period} * \text{current Timer2 Tick}.$$

5.1.5 Mode3 (Tick Mode)

In Mode 3, system clock is employed.

After Timer is enabled, Timer Tick starts counting upward, and Timer Tick value is increased by 1 on each positive edge of system clock.

This mode could be used as time indicator. There will be no interrupt generated. Timer Tick keeps rolling from 0 to 0xffffffff. When Timer tick overflows, it returnsto 0 and starts counting upward again.

Steps of setting Timer0for Mode 3 is taken as an example.

1st: Set initial Tick value of Timer0

Set Initial value of Tick viaaddress 0x630~0x633. Address 0x630 is lowest byte and address 0x633 is highest byte. It's recommended to clear initial Timer Tick value to 0.

2nd: Set Timer0 to Mode 3 and enable Timer0

Set address 0x620[2:1] to 2b'11 to select Mode 3, meanwhile set address 0x620[0] to 1b'1 to enable Timer0. Timer0 Tick starts to roll.

3rd: Read current Timer0 Tick value

Current Timer0 Tick value can be read from address 0x630~0x633.

5.1.6 Watchdog

Programmable watchdog could reset chip from unexpected hang up or malfunction.

Only Timer2 supports Watchdog.

Timer2 Tick has 32bits. Watchdog Capture has only 14bits, which consists of TMR_CTRL2 (address 0x622) [6:0] as higher bits and TMR_CTRL1 (address 0x621) [7:1] as lower bits. Chip will be reset when the Timer2 Tick[31:18] matches Watch dog capture.

1st: Clear Timer2 Tick value

Clear registers TMR_TICK2_0 ~TMR_TICK2_3 (address 0x638~0x63b). Address 0x638 is lowest byte and 0x63b is highest byte.

2nd:Enable Timer2

Set register TMR_CTRL0 (address 0x620) [6] to 1b'1 to enable Timer2.

3rd: Set 14-bitWatchdog Capture value and enable Watchdog

Set address 0x622[6:0] as higher bits of watchdog capture and 0x621[7:1] as lower bits. Meanwhile set address 0x622[7] to 1b'1 to enable Watchdog.

Then Timer2 Tick starts counting upwards from 0.

If bits[31:18] of Timer2 Tick value read from address 0x638~0x63b reaches watchdog capture, the chip will be reset, and the status bit in address 0x72[0] will be set as 1b'1 automatically. User can read the watchdog status bit after chip reset to check if the reset source is watchdog, and needs to write 1b'1 to this bit to manually clear the flag.

5.2 32K LTIMER

The ST17H58 also supports a low frequency (32KHz) LTIMER in suspend mode or deep sleep mode. This timer can be used as one kind of wakeup source.

5.3 System Timer

The ST57H25F512also supports a System Timer. As introduced in section **4.3.1 System Timer clock**, the clock frequency for System Timer is fixed as 16MHz irrespective of system clock.

In suspend mode, both System Timer and Timer0~Timer2 stop counting, and 32K Timer starts counting. When the chip restores to active mode, Timer0~Timer2 will continue counting from the number when theystops; In contrast, System Timer will continue counting from an adjusted number which is a sum of the number when it stops and an offset calculated from the counting value of 32K Timer during suspend mode.

Table 5- 2 Register table for System Timer

Address	Mnemonic	R/W	Function	Default Value
0x740	Sys_timer[7:0]	R/W		00
0x741	Sys_timer[15:8]	R/W		00
0x742	Sys_timer[23:16]	R/W		00
0x743	Sys_timer[31:24]	R/W	System timer counter, write to set initial value. This is the sys timer counter	00
0x74c	Sys_timer_ctrl	R/W	[7]:cal 32k enable (16 cycles 32k, count sys clock cycles) [6]:1:at the pos of 32k clock to set 32k timer value [5]:suspend bypass system_timer module [4]:system timer ss enable [3] manual set 32k timer mode [2]:manual set 32k timer 1:write,0: read [1]:irq mask, 1: enable, 0: disable [0] rsvd	0x90

***Note:** The lower four bits of address 0x740 is invalid, therefore, the resolution should be 1us.

6 Interrupt System

6.1 Interrupt structure

The interrupting function is applied to manage dynamic program sequencing based on real-time events triggered by timers, pins and etc.

For the ST17H58, there are 24 interrupt sources in all: 16 types are level-triggered interrupt sources (listed in address 0x640~0x641) and 8 types are edge-triggered interrupt sources (listed in address 0x642) .

WhenCPU receives an interrupt request (IRQ) from some interrupt source, it will determine whether to respond to the IRQ. If CPU decides to respond, it pauses current routine and starts to execute interrupt service subroutine. Program will jump to certain code address and execute IRQ commands. After finishing interrupt service subroutine, CPU returns to the breakpoint and continues to execute main function.

6.2 Register configuration

Table 6- 1 Register table for Interrupt system

Address	Mnemonic	Type	Description	Reset Value
0x640	MASK_0	RW	Byte 0 interrupt mask, level-triggered type {irq_qdec,irq_uart,irq_dfifo, irq_dma,usb_pwdn,time2,time1,time0} [7] irq_qdec [6] irq_uart [5] irq_dfifo [4]irq_dma [3] usb_pwdn [2] time2 [1] time1 [0] time0	00
0x641	MASK_1	RW	Byte 1 interrupt mask, level-triggered type {rsvd,irq_pwm,irq_zb_rt,irq_udc[4:0]} [7] rsvd [6] irq_pwm [5] irq_zb_rt [4] irq_udc[4] [3] irq_udc[3] [2] irq_udc[2] [1] irq_udc[1] [0] irq_udc[0]	00
0x642	MASK_2	RW	Byte 2 interrupt mask, edge-triggered type {irq_cmp,gpio2risc[1:0],irq_stimer,pm_irq,irq_ gpio,usb_reset,usb_250us} [7] irq_cmp [6] gpio2risc[1] [5] gpio2risc[0] [4] irq_stimer [3] pm_irq	00

Address	Mnemonic	Type	Description	Reset Value
			[2] irq_gpio [1] usb_reset [0] usb_250us	
0x643	IRQMODE	RW	[0] interrupt enable [1] reserved (Multi-Address enable)	00
0x644	PRIO_0	RW	Byte 0 of priority 1: High priority; 0: Low priority	00
0x645	PRIO_1	RW	Byte 1 of priority	00
0x646	PRIO_2	RW	Byte 2 of priority	00
0x648	IRQSRC_0	R	Byte 0 of interrupt source	
0x649	IRQSRC_1	R	Byte 1 of interrupt source	
0x64a	IRQSRC_2	R	Byte 2 of interrupt source	

6.2.1 Enable/Mask interrupt sources

Various interruptsources could be enabled or masked by registersMASK_0~MASK_2 (address 0x640~0x642).

6.2.2 Interrupt mode and priority

Interrupt mode is typically-used mode. Register IRQMODE (address 0x643)[0] should be set as 1b’1 to enable interrupt function.

IRQ tasks could be set as High or Low priority via registers PRIO_0~PRIO_2 (address 0x644~0x646). When more than one interrupt sources assert interrupt requests at the same time, CPU will respond depending on respective interrupt priority levels. It’s recommended not to modify priority setting.

6.2.3 Interrupt source flag

Three bytes in registers IRQSRC_0~IRQSRC_2 (address 0x648~0x64a) serve to indicate IRQ sources. Once IRQ occurs from certain source, the corresponding IRQ source flag will be set as “1”. User could identify IRQ source by reading address 0x648~0x64a.

When handling edge-triggered type interrupt, the corresponding IRQ source flag needs to be cleared via address 0x64a. Take the interrupt source usb_250us for example: First enable the interrupt source by setting address 0x642 bit[0] as 1b’1; then set address 0x643 bit[0] as 1b’1 to enable the interrupt. In interrupt handling function, 24-bit data is read from address 0x648~0x64a to check which IRQ source is valid; if data bit[16] is 1, it means the usb_250us IRQsourceis valid. Clear this interrupt source by setting address 0x64a bit[0] as 1b’1.

As for level-type interrupt, IRQ interrupt source status needs to be cleared via setting corresponding module status register. Take Timer0 IRQ interrupt source for example: First enable the interrupt source by setting address 0x640 bit[0] as 1b’1; then set address 0x643 bit[0] as 1b’1 to enable the interrupt. In interrupt handling function, 24-bit data is read from address 0x648~0x64a to check which IRQ source is valid; if data bit[0] is 1, it means the Timer0 IRQ source is valid. Register TMR_STATUS (address 0x623) [0] should be written with 1b’1 to manually clear Timer0 status (refer to section **5.1.1 Register table**).

7 Interface

7.1 GPIO

The ST17H58 support up to32, 17 and 10 GPIOs respectively. Except for dedicated GPIOs, all digital IOs can be used as general purpose IOs.

All GPIOs (including ANA_A<0>~ANA_D<7>)have configurable pull-up/pull-down resistor. Please refer to **section 7.1.3 Pull-up/Pull-down resistor** for details.

7.1.1 Basic configuration

Please refer to Table 7- 1in section 7.1.1.3 for various GPIO interface configuration.

7.1.1.1 Multiplexed functions

Each pin listed in Table 7-1 acts as the function in the "**Default Function**" column by default.

If a pin with multiplexed functions does not act as GPIO function by default, to use it as GPIO, first set the bit in "**Act as GPIO**" column as 1b'1. After GPIO function is enabled, if the pin is used as output, both the bits in "**IE**" and "**OEN**" columns should be set as 1b'0, then set the register value in the "**Output**" column; if the pin is used as input, both the bits in "**IE**" and "**OEN**" columns should be set as 1b'1, and the input data can be read from the register in the "**Input**" column.

To usea pin as certain multiplexed function (neither the default function nor GPIO function),firstclearthe bit in "**Act as GPIO**" column to disable GPIO function, and thenconfigure"**Register**" in "Pad Function Mux" column to enable multiplexed function correspondingly.

Example 1: DMIC_DI/ PWM0_N/ UART_RX/ ANA_A<0>.

(1) The pin acts as GPIO function by default.

- ✧ If the pin is used as general output, both address 0x581[0] (IE) and 0x582[0] (OEN) should be set as 1b'0, then configure address 0x583[0] (Output).
- ✧ If the pin is used as general input, both address 0x581[0] (IE) and 0x582[0] (OEN) should be set as 1b'1, and the input data can be read from address 0x580[0] (Input).

(2) To use the pin as DMIC_DI function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0](Register) should be set as 2b'00.

(3) To use the pin as PWM0_N function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0] (Register) should be set as 2b'01.

(4) To use the pin as UART_RX function, address 0x586[0] (Act as GPIO) should be set as 1b'0, and 0x5a8[1:0] (Register) should be set as 2b'10.

Example 2: SWS/ UART_RTS/ ANA_A<7>.

(1) The pin acts as SWS function by default.

(2) To use it as GPIO function, first set address 0x586[7] (Act as GPIO)as1b'1.

- ✧ If the pin is used as general output, both address 0x581[7] (IE) and 0x582[7] (OEN) should be set as 1b'0, then configure address 0x583[7] (Output).

✧ If the pin is used as general input, both address 0x581[7] (IE) and 0x582[7] (OEN) should be set to 1b'1, and the input data can be read from address 0x580[7] (Input).

(3) To use it as UART_RTSfunction, set address 0x586[7] (Act as GPIO) as 1b'0, and set 0x5a9[7:6] (Register) to 2b'01.

7.1.1.2 Drive strength

The registers in the "DS" column are used to configure the corresponding pin's driving strength: "1" indicates maximum drive level, while "0" indicates minimal drive level.

The "DS" configuration will take effect when the pin is used as output.It's set as the strongest driving level by default. In actual applications, driving strength can be decreased to lower level if necessary.

✧ ANA_A<0>~ANA_A<4>, ANA_C<6>~ANA_C<7>, ANA_D<0>~ANA_D<7>: maximum=4mA ("DS"=1), minimum=2mA ("DS"=0).

✧ ANA_A<5>~ANA_A<7>, ANA_B<0>~ANA_B<3>, ANA_C<0>~ANA_C<5>:maximum=8mA ("DS"=1), minimum=4mA ("DS"=0).

✧ ANA_B<4>~ANA_B<7>: maximum=16mA ("DS"=1), minimum=12mA ("DS"=0).

7.1.1.3 GPIO lookup table

Table 7- 1GPIO lookup table

Pin	Default function	Pad Function Mux					GPIO Setting						
		Register=3	Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	Act as GPIO
DMIC_DI/ PWM0_N/ UART_RX/ ANA_A<0>	GPIO	/	UART_RX	PWM0_N	DMIC_DI	0x5a8 [1:0]	0x580 [0]	0x581 [0]	0x582 [0]	0x583 [0]	0x584 [0]	0x585 [0]	0x586 [0]
DMIC_CLK/ 7816_CLK/ I2S_CLK/ ANA_A<1>	GPIO	/	I2S_CLK	UART_7816_CLK	DMIC_CLK	0x5a8 [3:2]	0x580 [1]	0x581 [1]	0x582 [1]	0x583 [1]	0x584 [1]	0x585 [1]	0x586 [1]
DO/ UART_TX/ ANA_A<2>	GPIO	/	PAD_CLKBB	UART_TX	DO	0x5a8 [5:4]	0x580 [2]	0x581 [2]	0x582 [2]	0x583 [2]	0x584 [2]	0x585 [2]	0x586 [2]
DI/ UART_CTS/ ANA_A<3>	GPIO	/	PAD_TX_EN	UART_CTS	DI	0x5a8 [7:6]	0x580 [3]	0x581 [3]	0x582 [3]	0x583 [3]	0x584 [3]	0x585 [3]	0x586 [3]
CK/ UART_RTS/ ANA_A<4>	GPIO	/	PAD_TX_CYC1	UART_RTS	CK	0x5a9 [1:0]	0x580 [4]	0x581 [4]	0x582 [4]	0x583 [4]	0x584 [4]	0x585 [4]	0x586 [4]
DM/ ANA_A<5>	GPIO	/	/	/	DM	0x5a9 [3:2]	0x580 [5]	0x581 [5]	0x582 [5]	0x583 [5]	0x584 [5]	0x585 [5]	0x586 [5]
DP/ ANA_A<6>	GPIO	/	/	/	DP	0x5a9 [5:4]	0x580 [6]	0x581 [6]	0x582 [6]	0x583 [6]	0x584 [6]	0x585 [6]	0x586 [6]

Pin	Default function	Pad Function Mux					GPIO Setting						
		Register=3	Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	Act as GPIO
SWS/ UART_RTS/ ANA_A<7>	SWS	/	/	UART_RTS	SWS	0x5a9 [7:6]	0x580 [7]	0x581 [7]	0x582 [7]	0x583 [7]	0x584 [7]	0x585 [7]	0x586 [7]
PWM0/ UART_TX/ PWM1_N/ ANA_B<0>	GPIO	/	PWM1_N	UART_TX	PWM0	0x5aa [1:0]	0x588 [0]	0xbd [0]	0x58a [0]	0x58b [0]	0x58c [0]	0xbf [0]	0x58e [0]
PWM1/ UART_RX/ PWM0_N/ ANA_B<1>	GPIO	/	PWM0_N	UART_RX	PWM1	0x5aa [3:2]	0x588 [1]	0xbd [1]	0x58a [1]	0x58b [1]	0x58c [1]	0xbf [1]	0x58e [1]
PWM2/ CMP_D/ RX_CYC2LNA/ ANA_B<2>	GPIO	/	RX_CYC2 LNA	CMP_D	PWM2	0x5aa [5:4]	0x588 [2]	0xbd [2]	0x58a [2]	0x58b [2]	0x58c [2]	0xbf [2]	0x58e [2]
PWM3/ CMP_DF/ TX_CYC2PA/ ANA_B<3>	GPIO	/	TX_CYC2PA	CMP_DF	PWM3	0x5aa [7:6]	0x588 [3]	0xbd [3]	0x58a [3]	0x58b [3]	0x58c [3]	0xbf [3]	0x58e [3]
SDM_P0/ PWM4/ PWM0_N/ ANA_B<4>	GPIO	/	PWM0_N	PWM4	SDM_P0	0x5ab [1:0]	0x588 [4]	0xbd [4]	0x58a [4]	0x58b [4]	0x58c [4]	0xbf [4]	0x58e [4]
SDM_N0/ PWM5/ PWM0/ ANA_B<5>	GPIO	/	PWM0	PWM5	SDM_N0	0x5ab [3:2]	0x588 [5]	0xbd [5]	0x58a [5]	0x58b [5]	0x58c [5]	0xbf [5]	0x58e [5]
SDM_P1/ PWM4_N/ UART_TX/ 7816_TRX/ ANA_B<6>	GPIO	/	UART_TX (7816)	PWM4_N	SDM_P1	0x5ab [5:4]	0x588 [6]	0xbd [6]	0x58a [6]	0x58b [6]	0x58c [6]	0xbf [6]	0x58e [6]
SDM_N1/ PWM5_N/ UART_RX/ ANA_B<7>	GPIO	/	UART_RX	PWM5_N	SDM_N1	0x5ab [7:6]	0x588 [7]	0xbd [7]	0x58a [7]	0x58b [7]	0x58c [7]	0xbf [7]	0x58e [7]
I2C_SDA/ PWM4/ PWM1/ PGA_P0/ ANA_C<0>	GPIO	/	PWM1	PWM4	I2C_SDA	0x5ac [1:0]	0x590 [0]	0xc0 [0]	0x592 [0]	0x593 [0]	0x594 [0]	0xc2 [0]	0x596 [0]
I2C_SCK/ PWM5/ PGA_N0/ ANA_C<1>	GPIO	/	PAD_RX_EN	PWM5	I2C_SCK	0x5ac [3:2]	0x590 [1]	0xc0 [1]	0x592 [1]	0x593 [1]	0x594 [1]	0xc2 [1]	0x596 [1]

Pin	Default function	Pad Function Mux					GPIO Setting						
		Register=3	Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	Act as GPIO
PWM0/ UART_RTS/ XC32K_O/ PGA_P1/ ANA_C<2>	GPIO	/	PAD_GAIN [0]	UART_RTS	PWM0	0x5ac [5:4]	0x590 [2]	0xc0 [2]	0x592 [2]	0x593 [2]	0x594 [2]	0xc2 [2]	0x596 [2]
PWM1_N/ UART_CTS/ XC32K_I/ PGA_N1/ ANA_C<3>	GPIO	/	PAD_GAIN [1]	UART_CTS	PWM1_N	0x5ac [7:6]	0x590 [3]	0xc0 [3]	0x592 [3]	0x593 [3]	0x594 [3]	0xc2 [3]	0x596 [3]
PWM2_N/ /UART_TX7816_TRX/ ANA_C<4>	GPIO	/	PAD_GAIN [2]	UART_TX (7816)	PWM2_N	0x5ad [1:0]	0x590 [4]	0xc0 [4]	0x592 [4]	0x593 [4]	0x594 [4]	0xc2 [4]	0x596 [4]
PWM3_N/ UART_RX/ ANA_C<5>	GPIO	/	PAD_GAIN [3]	UART_RX	PWM3_N	0x5ad [3:2]	0x590 [5]	0xc0 [5]	0x592 [5]	0x593 [5]	0x594 [5]	0xc2 [5]	0x596 [5]
RX_CYC2LNA/ CMP_D/ ANA_C<6>	GPIO	/	PAD_GAIN [4]	CMP_D	RX_CYC2 LNA	0x5ad [5:4]	0x590 [6]	0xc0 [6]	0x592 [6]	0x593 [6]	0x594 [6]	0xc2 [6]	0x596 [6]
TX_CYC2PA/ CMP_DF/ ANA_C<7>	GPIO	TRX_CLK	PAD_TRX_ CLK	CMP_DF	TX_CYC2 PA	0x5ad [7:6]	0x590 [7]	0xc0 [7]	0x592 [7]	0x593 [7]	0x594 [7]	0xc2 [7]	0x596 [7]
RX_CYC2LNA/ CMP_D/ UART_RTS/ PWM0/ ANA_D<0>	GPIO	PWM0	UART_RTS	CMP_D	RX_CYC2 LNA	0x5ae [1:0]	0x598 [0]	0x599 [0]	0x59a [0]	0x59b [0]	0x59c [0]	0x59d [0]	0x59e [0]
TX_CYC2PA/ CMP_DF/ UART_CTS/ PWM1/ ANA_D<1>	GPIO	PWM1	UART_CTS	CMP_DF	TX_CYC2 PA	0x5ae [3:2]	0x598 [1]	0x599 [1]	0x59a [1]	0x59b [1]	0x59c [1]	0x59d [1]	0x59e [1]
SPI_CN/ I2S_LR/ UART_RX/ PWM2/ ANA_D<2>	SPI_CN	PWM2	UART_RX	I2S_LR	SPI_CN	0x5ae [5:4]	0x598 [2]	0x599 [2]	0x59a [2]	0x59b [2]	0x59c [2]	0x59d [2]	0x59e [2]
SPI_DO/ I2S_SDI/ UART_TX/ 7816_TRX/ PWM3/ ANA_D<3>	SPI_DO	PWM3	UART_TX (7816)	I2S_SDI	SPI_DO	0x5ae [7:6]	0x598 [3]	0x599 [3]	0x59a [3]	0x59b [3]	0x59c [3]	0x59d [3]	0x59e [3]

Pin	Default function	Pad Function Mux					GPIO Setting						
		Register=3	Register=2	Register=1	Register=0	Register	Input (R)	IE	OEN	Output	Polarity	DS	Act as GPIO
SPI_CK/ I2S_SDO/ PWM0/ PWM4/ ANA_D<4>	SPI_CK	PWM4	PWM0	I2S_SDO	SPI_CK	0x5af [1:0]	0x598 [4]	0x599 [4]	0x59a [4]	0x59b [4]	0x59c [4]	0x59d [4]	0x59e [4]
SPI_DI/ I2S_BCK/ PWM0_N/ PWM5/ ANA_D<5>	SPI_DI	PWM5	PWM0_N	I2S_BCK	SPI_DI	0x5af [3:2]	0x598 [5]	0x599 [5]	0x59a [5]	0x59b [5]	0x59c [5]	0x59d [5]	0x59e [5]
CN/ UART_RX/ ANA_D<6>	GPIO	/	PAD_TX_SD	UART_RX	CN	0x5af [5:4]	0x598 [6]	0x599 [6]	0x59a [6]	0x59b [6]	0x59c [6]	0x59d [6]	0x59e [6]
SWM/ CMP_D/ UART_TX/ 7816_TRX/ ANA_D<7>	GPIO	/	UART_TX (7816)	CMP_D	SWM	0x5af [7:6]	0x598 [7]	0x599 [7]	0x59a [7]	0x59b [7]	0x59c [7]	0x59d [7]	0x59e [7]

***Notes:**

(1) IE: Input enable, high active. 1: enable input, 0: disable input.

(2) OEN: Output enable, low active. 0: enable output, 1: disable output.

(3) Register: Configure multiplexed functions in “Pad Function Mux” column.

(4) Output: configure GPO output.

(5) Input: read GPI input.

(6) DS: Drive strength. Default: 1.

(7) Act as GPIO: enable (1) or disable (0) GPIO function.

(8) Polarity: see section 7.1.2 Connection relationship between GPIO and related modules.

(9) Priority: “Act as GPIO”has the highest priority. To configure as multiplexed function, disable GPIO function first.

(10) 0xbd, 0xbf, 0xc0 and 0xc2 marked in red color are analog registers; others are digital registers.

7.1.2 Connection relationship between GPIO and related modules

GPIO can be used to generate GPIO interrupt signal for interrupt system,counting or control signal for Timer/Counter module, or GPIO2RISC interrupt signal for interrupt system.

For the “Exclusive Or (XOR)” operation result for input signal from any GPIO pin and respective “Polarity” value, on one hand, it takes “And” operation with “irq” and generates GPIO interrupt request signal; on the other hand, it takes “And” operation with “m0/m1/m2”, and generates counting signal in Mode 1 or control signal in Mode 2 for Timer0/Timer1/Timer2, or generates GPIO2RISC[0]/GPIO2RISC[1] interrupt request signal.

GPIO interrupt request signal = | ((input ^ polarity) &irq);

Counting (Mode 1) or control (Mode 2) signal for Timer0 = | ((input ^ polarity) & m0);

Counting (Mode 1) or control (Mode 2) signal for Timer1 = | ((input ^ polarity) & m1);

Counting (Mode 1) or control (Mode 2) signal for Timer2 = | ((input ^ polarity) & m2);

GPIO2RISC[0] interrupt request signal = | ((input ^ polarity) & m0);

GPIO2RISC[1] interrupt request signal = | ((input ^ polarity) & m1).

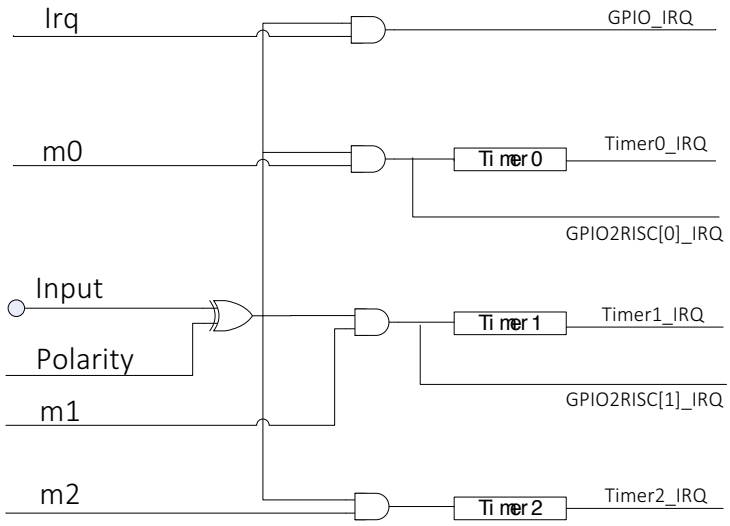


Figure 7- 1 Logic relationship between GPIO and related modules

Please refer to Table 7- 2and Table 6- 1to learn how to configure GPIO for interrupt system or Timer/Counter (Mode 1 or Mode 2).

(1) First enable GPIO function, enable IE and disable OEN. Please see section 7.1.1 Basic configuration.

(2) GPIO IRQ signal:

Select GPIO interrupt trigger edge (positive edge or negative edge) via configuring "**Polarity**", and set corresponding GPIO interrupt enabling bit "**Irq**".

Then set address 0x5b5[3] to enable GPIO IRQ.

Finally enable GPIO interrupt (irq_gpio)viaaddress 0x642[2]).

User can read addresses 0x5e0 ~ 0x5e3 to see which GPIO asserts GPIO interrupt request signal.Note: 0x5e0[7:0] --> ANA_A<7>~ANA_A<0>, 0x5e1[7:0] --> ANA_B<7>~ANA_B<0>, 0x5e2[7:0] --> ANA_C<7>~ANA_C<0>, 0x5e3[7:0] --> ANA_D<7>~ANA_D<0>.

(3) Timer/Counter counting or control signal:

Configure "**Polarity**". In Timer Mode 1，it determines GPIO edge when Timer Tick counting increases.InTimerMode 2, it determines GPIO edge when Timer Tick starts counting.

Then set "**m0/m1/m2**" to specify the GPIO which generates counting signal (Mode1)/control signal(Mode2) for Timer0/Timer1/Timer2.

User can read addresses 0x5e8~0x5eb/0x5f0~0x5f3/0x5f8~0x5fb to see which GPIO asserts counting signal (in Mode 1) or control signal (in Mode 2) for Timer0/Timer1/Timer2. Note: Timer0: 0x5e8[7:0] --> ANA_A<7>~ANA_A<0>, 0x5e9[7:0] --> ANA_B<7>~ANA_B<0>, 0x5ea[7:0] --> ANA_C<7>~ANA_C<0>, 0x5eb[7:0] --> ANA_D<7>~ANA_D<0>; Timer1: 0x5f0[7:0] --> ANA_A<7>~ANA_A<0>, 0x5f1[7:0] --> ANA_B<7>~ANA_B<0>, 0x5f2[7:0] --> ANA_C<7>~ANA_C<0>, 0x5f3[7:0] --> ANA_D<7>~ANA_D<0>; Timer2: 0x5f8[7:0] --> ANA_A<7>~ANA_A<0>, 0x5f9[7:0] --> ANA_B<7>~ANA_B<0>, 0x5fa[7:0] --> ANA_C<7>~ANA_C<0>, 0x5fb[7:0] --> ANA_D<7>~ANA_D<0>.

(4) GPIO2RISC IRQ signal:

Select GPIO2RISC interrupt trigger edge (positive edge or negative edge) via configuring "**Polarity**", and set corresponding GPIO enabling bit **"m0"/"m1"**.

Enable GPIO2RISC[0]/GPIO2RISC[1] interrupt, i.e. "gpio2risc[0]" (address 0x642[5]) / "gpio2risc[1]"(address 0x642[6]).

Table 7- 2 GPIO lookup table2

Pin	Input (R)	Polarity 1: active low 0: active high	Irq	m0	m1	m2
ANA_A<0>	0x580[0]	0x584[0]	0x587[0]	0x5b8[0]	0x5c0[0]	0x5c8[0]
ANA_A<1>	0x580[1]	0x584[1]	0x587[1]	0x5b8[1]	0x5c0[1]	0x5c8[1]
ANA_A<2>	0x580[2]	0x584[2]	0x587[2]	0x5b8[2]	0x5c0[2]	0x5c8[2]
ANA_A<3>	0x580[3]	0x584[3]	0x587[3]	0x5b8[3]	0x5c0[3]	0x5c8[3]
ANA_A<4>	0x580[4]	0x584[4]	0x587[4]	0x5b8[4]	0x5c0[4]	0x5c8[4]
ANA_A<5>	0x580[5]	0x584[5]	0x587[5]	0x5b8[5]	0x5c0[5]	0x5c8[5]
ANA_A<6>	0x580[6]	0x584[6]	0x587[6]	0x5b8[6]	0x5c0[6]	0x5c8[6]
ANA_A<7>	0x580[7]	0x584[7]	0x587[7]	0x5b8[7]	0x5c0[7]	0x5c8[7]
ANA_B<0>	0x588[0]	0x58c[0]	0x58f[0]	0x5b9[0]	0x5c1[0]	0x5c9[0]
ANA_B<1>	0x588[1]	0x58c[1]	0x58f[1]	0x5b9[1]	0x5c1[1]	0x5c9[1]
ANA_B<2>	0x588[2]	0x58c[2]	0x58f[2]	0x5b9[2]	0x5c1[2]	0x5c9[2]
ANA_B<3>	0x588[3]	0x58c[3]	0x58f[3]	0x5b9[3]	0x5c1[3]	0x5c9[3]
ANA_B<4>	0x588[4]	0x58c[4]	0x58f[4]	0x5b9[4]	0x5c1[4]	0x5c9[4]
ANA_B<5>	0x588[5]	0x58c[5]	0x58f[5]	0x5b9[5]	0x5c1[5]	0x5c9[5]
ANA_B<6>	0x588[6]	0x58c[6]	0x58f[6]	0x5b9[6]	0x5c1[6]	0x5c9[6]
ANA_B<7>	0x588[7]	0x58c[7]	0x58f[7]	0x5b9[7]	0x5c1[7]	0x5c9[7]
ANA_C<0>	0x590[0]	0x594[0]	0x597[0]	0x5ba[0]	0x5c2[0]	0x5ca[0]
ANA_C<1>	0x590[1]	0x594[1]	0x597[1]	0x5ba[1]	0x5c2[1]	0x5ca[1]
ANA_C<2>	0x590[2]	0x594[2]	0x597[2]	0x5ba[2]	0x5c2[2]	0x5ca[2]
ANA_C<3>	0x590[3]	0x594[3]	0x597[3]	0x5ba[3]	0x5c2[3]	0x5ca[3]
ANA_C<4>	0x590[4]	0x594[4]	0x597[4]	0x5ba[4]	0x5c2[4]	0x5ca[4]
ANA_C<5>	0x590[5]	0x594[5]	0x597[5]	0x5ba[5]	0x5c2[5]	0x5ca[5]
ANA_C<6>	0x590[6]	0x594[6]	0x597[6]	0x5ba[6]	0x5c2[6]	0x5ca[6]
ANA_C<7>	0x590[7]	0x594[7]	0x597[7]	0x5ba[7]	0x5c2[7]	0x5ca[7]

Pin	Input (R)	Polarity 1: active low 0: active high	Irq	m0	m1	m2
ANA_D<0>	0x598[0]	0x59c[0]	0x59f[0]	0x5bb[0]	0x5c3[0]	0x5cb[0]
ANA_D<1>	0x598[1]	0x59c[1]	0x59f[1]	0x5bb[1]	0x5c3[1]	0x5cb[1]
ANA_D<2>	0x598[2]	0x59c[2]	0x59f[2]	0x5bb[2]	0x5c3[2]	0x5cb[2]
ANA_D<3>	0x598[3]	0x59c[3]	0x59f[3]	0x5bb[3]	0x5c3[3]	0x5cb[3]
ANA_D<4>	0x598[4]	0x59c[4]	0x59f[4]	0x5bb[4]	0x5c3[4]	0x5cb[4]
ANA_D<5>	0x598[5]	0x59c[5]	0x59f[5]	0x5bb[5]	0x5c3[5]	0x5cb[5]
ANA_D<6>	0x598[6]	0x59c[6]	0x59f[6]	0x5bb[6]	0x5c3[6]	0x5cb[6]
ANA_D<7>	0x598[7]	0x59c[7]	0x59f[7]	0x5bb[7]	0x5c3[7]	0x5cb[7]

7.1.3 Pull-up/Pull-down resistor

All GPIOs (including ANA_A<0>~ANA_D<7>)support configurable 1MΩ/10KΩ pull-up resistor or 100KΩ pull-down resistor which are all disabled by default. Analog registers afe3V_reg14<7:0>~afe3V_reg21<7:0>serve to control the pull-up/pull-down resistor for each GPIO.

The DPpinalso supports 1.5KΩ pull-up resistor for USB use. The 1.5KΩ pull up resistor is disabled by default and can be enabled by settinganalog register afe3V_reg11<7> as 1b'1.For the DP/ANA_A<6> pin, user can only enable either 1.5KΩ pull up or 1MΩ/10KΩ pull-up/ 100KΩ pull-down resistor at the same time.

Please refer to Table 7- 3for details.

Take the ANA_A<0> for example: Setting analog register afe3V_reg14<1:0> to 2b'01/2b'11/2b'10 is to enable 1MΩ pull-up resistor/10KΩ pull-up resistor/100KΩ pull-down resistor respectively for ANA_A<0>; Clearing the two bits (default value) disables pull-up and pull-down resistor for ANA_A<0>.

Table 7- 3 Analog registers for pull-up/pull-down resistor control

Address	Mnemonic	Default	Description
afe3V_reg11<7>	dp_pullup_res_3v	0	1.5K pull-up resistor for USB DP PAD 0: disable 1: enable
afe3V_reg14<7:0>	a_sel<7:0>	00000000	ANA_A<3:0> pull up and down select: <7:6>: ANA_A<3> <5:4>: ANA_A<2> <3:2>: ANA_A<1> <1:0>: ANA_A<0> 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up

Address	Mnemonic	Default	Description
afe3V_reg15<7:0>	a_sel<15:8>	00000000	ANA_A<7:4> pull up and down select: <7:6>: ANA_A<7> <5:4>: ANA_A<6> <3:2>: ANA_A<5> <1:0>: ANA_A<4> 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg16<7:0>	b_sel<7:0>	00000000	ANA_B<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg17<7:0>	b_sel<15:8>	00000000	ANA_B<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg18<7:0>	c_sel<7:0>	00000000	ANA_C<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg19<7:0>	c_sel<15:8>	00000000	ANA_C<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg20<7:0>	d_sel<7:0>	00000000	ANA_D<3:0> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up
afe3V_reg21<7:0>	d_sel<15:8>	00000000	ANA_D<7:4> pull up and down select: 00: Null 01: 1M pull up 10: 100K pull down 11: 10K pull up

7.2 SWM and SWS

The ST17H58 supports Single Wire interface. SWM (Single Wire Master) and SWS (Single Wire Slave) represent the master and slave device of the single wire communication system developed by Lenze. The maximum data rate can be up to 2Mbps.

7.3 I2C

The ST57H25F512embeds I2C hardware module, which could act as Master mode or Slave mode. I2C is a popular inter-IC interface requiring only 2 bus lines, a serial data line (SDA) and a serial clock line (SCL).

7.3.1 Communication protocol

Lenze I2C module supports standard mode (100kbps), Fast-mode (400kbps)andFast-mode plus (1Mbps) with restriction that system clock must be by at least 10x of data rate.

Two wires, SDA and SCL (SCK) carry information between Master device and Slave device connected to the bus. Each device is recognized by unique address (ID). Master device is the device which initiates a data transfer on the bus and generates the clock signals to permit that transfer. Slave device is the device addressed by a Master.

Both SDA and SCL are bidirectional lines connected to a positive supply voltage via a pull-up resister. When the bus is free, both lines are HIGH.It's noted that data in SDA line must keep stable when clock signal in SCL line is at high level, and level state in SDA line is only allowed to change when clock signal in SCL line is at low level.

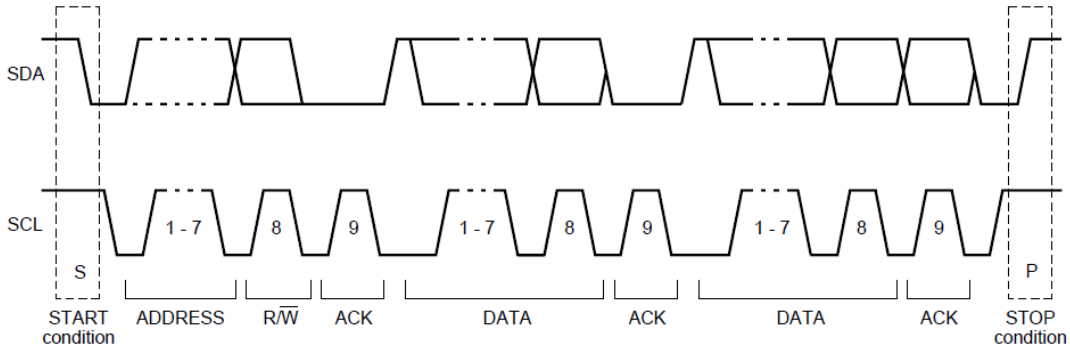


Figure 7- 2 I2C timing chart

7.3.2 Register table

Table 7- 4 Register configuration for I2C

Address	Name	R/W	Description	Reset Value
0x00	I2CSP	RW	I2C master clock speed	0x1f
0x01	I2CID	RW	[7:1]I2C ID	0x5c
0x02	I2CMST	RW	[0]: master busy [1]: master packet busy [2]: master received status 0 for ACK; 1 for NAK	
0x03	I2CSCT	RW	[0]: address auto increase enable [1]: I2C master enable [2] enable Mapping Mode	0x01
0x04	I2CAD	RW	[7:0] data buffer in master mode	0x5a
0x05	I2CDW	RW	[7:0] Data buffer in master mode	0xf1
0x06	I2CDR	RW	[7:0] Data buffer for Read or Write in	0x00

Address	Name	R/W	Description	Reset Value
			master mode	
0x07	I2CCLT	RW	[0]: launch ID cycle [1]: launch address cycle (send I2CAD data) [2]: launch data write cycle [3]: launch data read cycle For Master Write: 0: I2CAD&I2CDW, 1: I2CAD&I2CDW&I2CDR) To write 3 bytes: bit[3]=1; To write 2 bytes: bit[3]=0. For Master Read: always 1. [4]: launch start cycle [5]: launch stop cycle [6]: enable read ID [7]: enable ACK in read command	0x00
0xe0	Reg_host_map_status	R	[6:0] I2C read address	0x00
0xe1	Reg_host_map_adrl	R/W	Low byte of Mapping mode buffer address	0x80
0xe2	Reg_host_map_adrm	R/W	Middle byte of Mapping mode buffer address	0xd7
0xe3	Reg_host_map_adrh	R/W	High byte of Mapping mode buffer address	0x00
0xe4	i2c_status		[0]:host_cmd_irq_o i2c host operation have happened. Write 1 to clear. [1]:host_rd_tag_stat i2c host operation have happened and is read operation. Write 1 to clear.	0x00

7.3.3 I2C Slave mode

I2C module of the ST17H58 acts as Slave mode by default.I2Cslave address can be configured via register I2CID (address 0x01) [7:1].

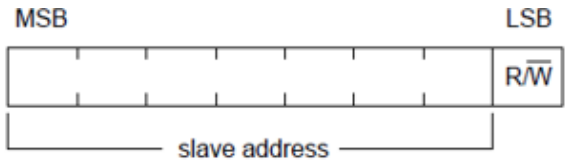


Figure 7- 3 Byte consisted of slave address and R/W flag bit

I2C slave mode supports two sub modes including Direct Memory Access (DMA) mode and Mapping mode, which is selectable via address 0x03[2].

In I2C Slave mode, Master could initiate transaction anytime. I2C slave module will reply with ACK automatically. To monitor the start of I2C transaction, user could set interrupt from GPIO for SCA or SCL.

7.3.3.1 DMA mode

In DMA mode, other devices (Master) could access (read/write) designated address in Register and/or SRAM of the ST17H58 according to I2C protocol. I2C module of the ST57HF512 will execute the read/write command from I2C master automatically. But user needs to notice that the system clock shall be at least 10x faster than I2C bit rate.

The access address designated by Master is offset by 0x800000. In the ST17H58, Register address starts from 0x800000 and SRAM address starts from 0x840000. For example, if Addr High(AddrH) is 0x04, Addr Middle (AddrM) is 0x00, and Addr Low (AddrL) is 0xcc, the real address of accessed data is 0x8400cc.

In DMA mode, Master could read/write data byte by byte. The designated access address is initial address and it supports auto increment by setting address 0x03[0] to 1b'1.

Read Format in DMA mode

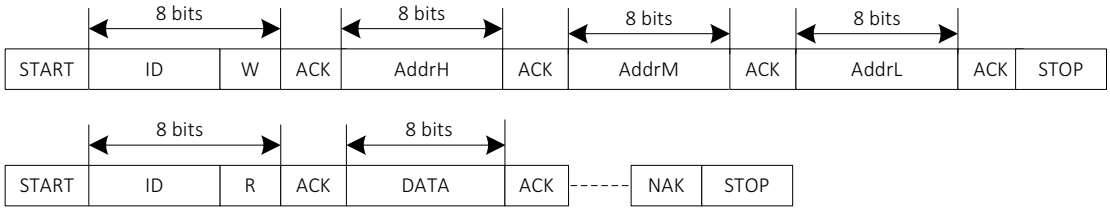


Figure 7- 4 Read format in DMA mode

Write Format in DMA mode

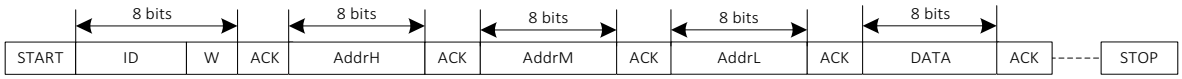


Figure 7- 5 Write format in DMA mode

7.3.3.2 Mapping mode

Mapping mode could be enabled via setting register I2CSCT (address 0x03)[2] to 1b'1.

In Mapping mode, data written and read by I2C master will be redirected to specified 128-byte buffer in SRAM. User could specify the initial address of the buffer by configuring registers reg_host_map_adrl (address 0xe1, lower byte), reg_host_map_adrm(address 0xe2, middle byte) and reg_host_map_adrh (address 0xe3, higher byte). The first 64-byte buffer is for written data and following 64-byte buffer is for read data. Every time the data access will start from the beginning of the Write-buffer/Read-buffer after I2C stop condition occurs. The last accessed data address could be checked in register reg_host_map_status (address 0xe0) [6:0] which is only updated after I2C STOP occurs.

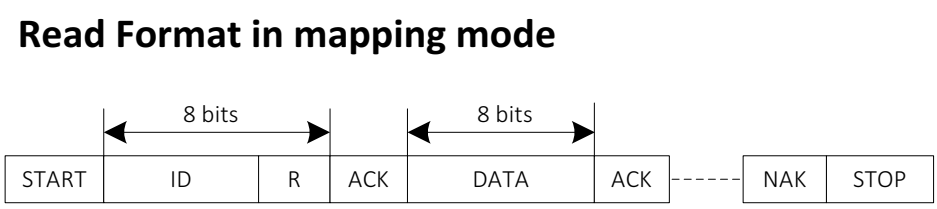


Figure 7- 6 Read format in Mapping mode

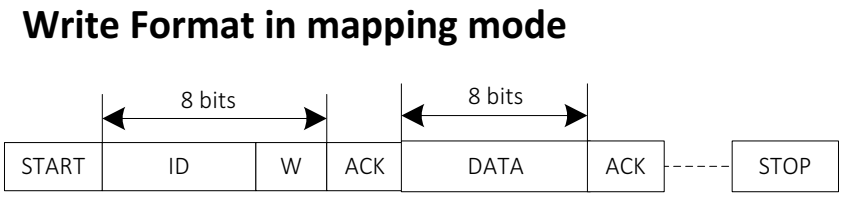


Figure 7- 7 Write format in Mapping mode

7.3.4 I2C Master mode

Address 0x03[1] should be set to 1b'1 to enable I2C master mode for the ST17H58. Address 0x00 serves to set I2C Master clock:F_{I2C} = (System Clock/(address 0x73[7:4]+1)) / (4 *clock speed configured in address 0x00). Since address 0x73[7:4] is set as 1 by default, the default F_{I2C} equals System Clock / (8 * address 0x00). If 0x73[7:4] is set as 0, F_{I2C} will change to System Clock / (4 * address 0x00).

A complete I2C protocol contains START, Slave Address, R/W bit, data, ACK and STOP. Slave address could be configured via address 0x01[7:1].

I2C Master(i.e. I2C module of the ST57HF512) could sendSTART, Slave Address, R/W bit, data and STOP cycle by configuring address 0x07. I2C master will send enabled cycles in the correct sequence.

Address 0x02 serves to indicate whether Master/Master packet is busy, as well as Master received status. Bit[0] will be set to 1 when one byte is being sent, and the bit can be automatically cleared after a start signal/ address byte/acknowledge signal/data /stop signal is sent. Bit[1] is set to 1 when the start signal is sent, and the bit will be automatically cleared after the stop signal is sent. Bit[2] indicates whether to succeed in sending acknowledgement signal.

7.3.4.1 I2C Master Write transfer

I2C Master has 3-byte buffer for write data, which are I2CAD (0x04), I2CDW (0x05) and I2CDR (0x06). Write transfer will be completed by I2C master module.

For example, to implement an I2C write transfer with 3-byte data, which contains START, Slave Address, Write bit, ack from Slave, 1st byte, ack from slave, 2nd byte, ack from slave, 3rd byte, ack from slave and STOP, user needs to configure I2C slave address to I2CID (0x01) [7:1], 1st byte data to I2CAD, 2nd byte data to I2CDW and 3rd byte to I2CDR. To start I2C write transfer, I2CCLT (0x07) is configured to 0x3f(0011 1111). I2C Master will launch START, Slave address, Write bit, load ACK

to I2CMST (0x02) [2], send I2CAD data, load ACK to I2CMST[2], send I2CDW data, load ACK to I2CMST[2], send I2CDR data, load ACK to I2CMST[2] and then STOP sequentially.

For I2C write transfer whose data are more than 3 bytes, user could split the cycles according to I2C protocol.

7.3.4.2 I2C Master Read transfer

I2C Master has one byte buffer for read data, which is I2CDR (0x06). Read transfer will be completed by I2C Master.

For example, to implement an I2C read transfer with 1 byte data, which contains START, Slave Address, Read bit, Ack from Slave, 1st byte from Slave, Ack by master and STOP, user needs to configure I2C slave address to I2CID (0x01) [7:1]. To start I2C read transfer, I2CCLT (0x07) is configured to 0xf9 (1111 1001). I2C Master will launch START, Slave address, Read bit, load ACK to I2CMST (0x02) [2], load data to I2CDR, reply ACK and then STOP sequentially.

For I2C read transfer whose data are more than 1 byte, user could split the cycles according to I2C protocol.

7.3.5 I2C and SPI Usage

I2C hardware and SPI hardware modules in the chip share part of the hardware, as a result, when both hardware interfaces are used, the restrictions listed within this section need to be taken into consideration.

I2C and SPI hardware cannot be used as Slave at the same time. I2C Slave mode and SPI Master mode cannot be used at the same time. I2C Master mode and SPI Slave mode cannot be used at the same time.

I2C and SPI can be used as Master at the same time only when specific GPIO pins are configured as I2C and SPI pins, respectively. Please refer to corresponding SDK instructions for details.

7.4 SPI

The ST57H25F512embeds SPI (Serial Peripheral interface), which could act as Master mode or Slave mode. SPI is a high-speed, full-duplex and synchronouscommunication bus requiring 4 bus lines including a chip select (CS) line, a data input (DI) line, a data output (DO) line and a clock (CK) line.

7.4.1 Register table

Table 7- 5 Register configuration for SPI

Address	Name	R/W	Description	Reset Value
0x08	SPIDAT	RW	SPI data access	
0x09	SPICT	RW	[0]: p_csn [1]: enable master mode [2]: spi data output disable [3]: 1 for read command; 0 for write	11

Address	Name	R/W	Description	Reset Value
			command [4]: address auto increase [5]: share_mode [6]: busy status	
0x0a	SPISP	RW	[6:0]: SPI clock speed [7]: SPI function mode, p_csn, p_scl, p_sda and p_sdo function as SPI if 1	05
0x0b	SPIMODE	RW	[0]: inverse SPI clock output [1]: dat delay half clk	0

7.4.2 SPI Master mode

SPI for the ST17H58 supports both master mode and slave mode and acts as slave mode by default. Address 0x09 bit[1] should be set to 1b'1 to enable SPI Master mode.Register SPISP is to configure SPI pin and clock: setting address 0x0a bit[7] to 1 is to enable SPI function mode, and corresponding pins can be used as SPI pins; SPI clock = system clock/((clock speed configured in address 0x0a bit[6:0] +1) *2) .

Address 0x08 serves as the data register. One reading/writing operation of 0x08 enables the SPI_CK pin to generate 8 SPI clock cycles.

Lenze SPI supports four standard working modes: Mode 0~Mode 3. Register SPIMODE (address 0x0b) serves to select one of the four SPI modes:

Table 7- 6 SPI mode

SPI mode	CPOL/CPHA	SPIMODE register (Address 0x0b)
Mode 0	CPOL=0, CPHA=0	bit[0]=0, bit[1]=0
Mode 1	CPOL=0, CPHA=1	bit[0]=0, bit[1]=1
Mode 2	CPOL=1, CPHA=0	bit[0]=1, bit[1]=0
Mode 3	CPOL=1, CPHA=1	bit[0]=1, bit[1]=1
CPOL: Clock Polarity When CPOL=0, SPI_CLK keeps low level in idle state; When CPOL=1, SPI_CLK keeps high level in idle state. CPHA: Clock Phase When CPHA=0, data is sampled at the first edge of clock period When CPHA=1, data is sampled at the latter edge of clock period		

Address 0x09 bit[0] is to control the CS line: when the bit is set to 1, the CS level is high; when the bit is cleared, the CS level is low.

Address 0x09bit[2] is the disabling bit for SPI Master output.When the bit is cleared, MCU writes data into address 0x08, then the SPI_DO pin outputs the data bit by bit during the 8 clock cycles generated by the SPI_CK pin. When the bit is set to 1b'1, SPI_DO output is disabled.

Address 0x09bit[3] is the enabling bit for SPI Master reading data function. When the bit is set to 1b'1, MCU reads the data from address 0x08, then the input data from the SPI_DI pin is shifted into address 0x08 during the 8 clock cycles generated by the SPI_CK pin. When the bit is cleared, SPI Master readingfunctionis disabled.

Address 0x09[5] is the enabling bit for share mode, i.e. whether SPI_DI and SPI_DO share one common line.

Users can read address 0x09 bit[6] to get SPI busy status, i.e. whether the 8 clock pulses have been sent.

7.4.3 SPI Slave mode

SPI for the ST17H58 acts as slave mode by default. SPI Slave mode supports DMA. User could access registers of the ST17H58 by SPI interface. It's noted that system clock of ST17H58 shall be at least 5x faster than SPI clock for reliable connection.Address0x0a should be written with data 0xa5 by the SPI host to activate SPI slave mode.

Address 0x09[4] is dedicated for SPI Slave mode and indicates address auto increment. SPI write command format and read command format are illustrated in Figure 7-8:

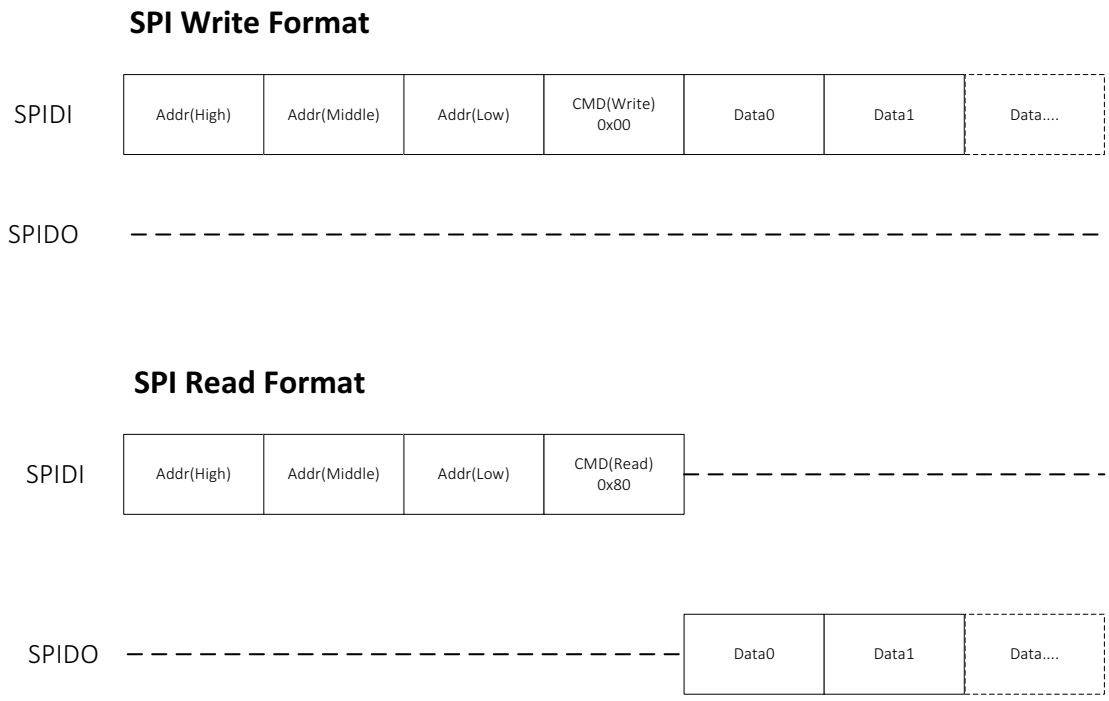


Figure 7- 8 SPI write/read command format

7.4.4 I2C and SPI Usage

I2C hardware and SPI hardware modules in the chip share part of the hardware, as a result, when both hardware interfaces are used, certain restrictions apply. See **Section7.3.5 I2C and SPI Usage** for detailed instructions.

7.5 UART

The ST57H25F512embeds UART (Universal Asynchronous Receiver/Transmitter) to implement full-duplex transmission and reception via UART TX and RX interface. Both TX and RX

interface are 4-layer FIFO (First In First Out) interface.

Hardware flow control is supported via RTS and CTS.

The UART module also supports ISO7816 protocol to enable communication with ISO/IEC 7816 integrated circuit card, especially smart card. In this mode, half-duplex communication (transmission or reception) is supported via the shared 7816_TRX interface.

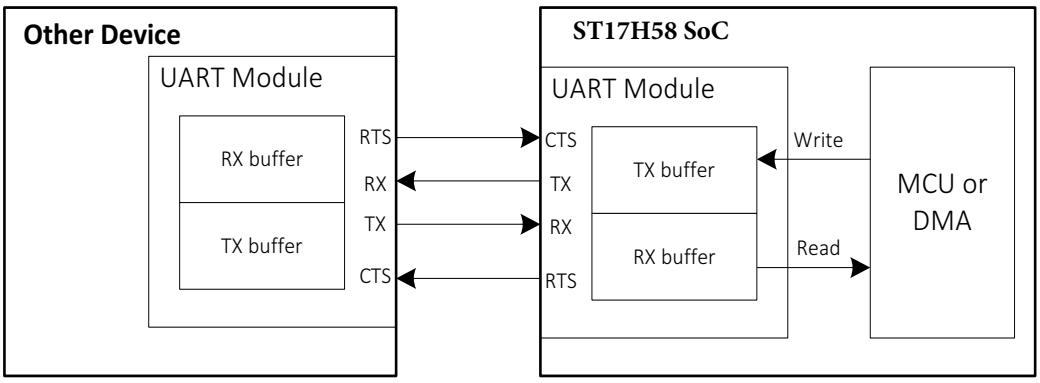


Figure 7- 9 UART communication

As shown in Figure 7-9, data to be sent is first written into TX buffer byMCU or DMA, then UART module transmits the data from TX buffer to other device via pin TX. Data to be read from other device is first received via pin RX and sent to RX buffer, then the data is read by MCU or DMA.

If RX buffer of the ST17H58 UART is close to full, the ST17H58 will send a signal (configurable high or low level) via pin RTS to inform other device that it should stop sending data. Similarly, if the ST17H58 receives a signal from pin CTS, it indicates that RX buffer of other device is close to full and the ST17H58 should stop sending data.

Table 7- 7 Register configuration for UART

Address	Name	R/W	Description	Reset Value
0x90	uart_data_buf0	R/W	write/read buffer[7:0]	
0x91	Uart_data_buf1	R/W	Write/read buffer[15:8]	
0x92	Uart_data_buf2	RW	Write/read buffer[23:16]	
0x93	Uart_data_buf3	R/W	Write/read buffer[31:24]	
0x94	uart_clk_div[7:0]	RW	uartclk div register:	0xff
0x95	Uart_clk_div[15:8]	R/W	uart_sclk = sclk/(uart_clk_div[14:0]+1) uart_clk_div[15] : 1:enable clock divider,0: disable.	0x0f
0x96	Uart ctrl0	R/W	[3:0] bwpc, bit width, should be larger than 2 Baudrate = uart_sclk/(bwpc+1) [4] rxdma enable [5] txdma enable [6] rx interrupt enable [7]tx interrupt enable	0x0f

Address	Name	R/W	Description	Reset Value
0x97	Uart_ctrl1	R/W	[0] cts select, 0: cts_i, 1: cts _i inverter [1]:cts enable, 1: enable, 0, disable [2]:Parity, 1: enable, 0 :disable [3]: even Parity or odd [5:4]: stop bit 00: 1 bit, 01, 1.5bit 1x: 2bits [6]: ttl [7]: uarttx, rx loopback	0x0e
0x98	Uart_ctrl2	R/W	[3:0] rts trig level [4] rtsParity [5] rts manual value [6] rts manual enable [7] rts enable	0xa5
0x99	Uart_ctrl3	R/W	[3:0]: rx_irq_trig level [7:4] tx_irq_trig level	0x44
0x9a	R_rxtimeout_o[7:0]	R/W	The setting is transfer one bytes need cycles base on uart_clk. For example, if transfer one bytes (1start bit+8bits data+1 priority bit+2stop bits) total 12 bits, this register setting should be (bwpc+1)*12.	0x0f
0x9b	R_rxtimeout_o[9:8]	R/W	2'b00:rx timeout time is r_rxtimeout[7:0] 2'b01:rx timeout time is r_rxtimeout[7:0]*2 2'b10:rx timeout time is r_rxtimeout[7:0]*3 3'b11: rx timeout time is r_rxtimeout[7:0]*4 R_rxtimeout is for rxdma to decide the end of each transaction. Supposed the interval between each byte in one transaction is very short.	0x00
0x9c	Buf_cnt	R	[3:0]: r_buf_cnt [7:4]:t_buf_cnt	
0x9d	Uart_sts	R	[2:0] rbcnt [3] irq [6:4]wbcnt [6] write 1 clear rx [7] rx_err, write 1 clear tx	

Addresses 0x90~0x93 serve to write data into TX buffer or read data from RX buffer.

Addresses 0x94~0x95 serve to configure UART clock.

Address 0x96 serves to set baud rate (bit[3:0]), enable RX/TX DMA mode (bit[4:5]), and enable RX/TX interrupt (bit[6:7]).

Address 0x97 mainly serves to configure CTS. Bit[1] should be set to 1b'1 to enable CTS. Bit[0]

serves to configure CTS signal level. Bit[2:3] serve to enable parity bit and select even/odd parity. Bit[5:4] serve to select 1/1.5/2 bits for stop bit. Bit[6] serves to configure whether RX/TX level should be inverted.

Address 0x98 serves to configure RTS. Bit[7] and Bit[3:0] serve to enable RTS and configure RTS signal level.

Address 0x99 serves to configure the number of bytes in RX/TX buffer to trigger interrupt.

The number of bytes in RX/TX buffer can be read from address 0x9c.

7.6 USB

The ST17H58 has a full-speed (12Mbps) USB interface for communicating with other compatible digital devices. The USB interface acts as a USB peripheral, responding to requests from a master host controller. The chip contains internal 1.5kohm pull up resistor for the DP pin, which can be enabled via analog registerafe3V_reg11<7>.

Lenze USB interfacesupports the Universal Serial Bus Specification, Revision v2.0 (USB v2.0 Specification).

The chip supports 9 endpoints, including control endpoint 0 and 8 configurable data endpoints. Endpoint 1, 2,3,4,7 and 8 can be configured as input endpoint, while endpoint 5 and 6 can be configured as output endpoint. In audio class application, only endpoint 6 supportsiso out mode, while endpoint 7 supportsiso in mode. In other applications, each endpoint can be configured as bulk, interrupt and iso mode. For control endpoint 0, the chip's hardware vendor command is configurable.

Optional suspend mode:

- ✧ Selectable as USB suspend mode or chip suspend mode, support remote wakeup.
- ✧ Current draw in suspend mode complied with USB v2.0 Specification.
- ✧ USB pins (DM, DP) can be used as GPIO function in suspend mode.
- ✧ Resume and detach detect: Recognize USB device by detecting the voltage on the DP pin with configurable 1.5K pull-up resistor.
- ✧ USB pins configurable as wakeup GPIOs.

The USB interface belongs to an independent power domain, and it can be configured to power down independently.

8 PWM

The ST17H58 supports 6-channel PWM (Pulse-Width-Modulation) output. Each PWM#n(n=0~5) has its corresponding inverted output at PWM#n_Npin.

8.1 Register table

Table 8- 1 Register table for PWM

Address	Mnemonic	Type	Description	Reset Value
0x780	PWM_EN	R/W	[0]: 0--disable PWM0, 1--enable PWM0 [1]: 0--disable PWM1, 1--enable PWM1 [2]: 0--disable PWM2, 1--enable PWM2 [3]: 0--disable PWM3, 1--enable PWM3 [4]: 0--disable PWM4, 1--enable PWM4 [5]: 0--disable PWM5, 1--enable PWM5	0x00
0x781	PWM_CLK	R/W	(PWM_CLK+1)*sys_clk	0x00
0x782	PWM_MODE	R/W	[1:0]: 00-pwm0 normal mode [1:0]: 01-pwm0 count mode [1:0]: 11-pwm0 IR mode [4][1:0]:111-pwm0 IR FIFO mode	0x00
0x783	PWM_CC0	R/W	[5:0]:1‘b1 invert PWM output	0x00
0x784	PWM_CC1	R/W	[5:0]:1‘b1 invert PWM_INV output	0x00
0x785	PWM_CC2	R/W	[5:0]: Signal frame polarity of PWM5~PWM0 1b’0-high level first 1b’1-low level first	0x00
0x788	PWM_PHASE0	R/W	[7:0] bits 7-0 of PWM0's phase time	0x00
0x789	PWM_PHASE0	R/W	[15:8] bits 15-8 of PWM0's phase time	0x00
0x78a	PWM_PHASE1	R/W	[7:0] bits 7-0 of PWM1's phase time	0x00
0x78b	PWM_PHASE1	R/W	[7:8] bits 15-8 of PWM1's phase time	0x00
0x78c	PWM_PHASE2	R/W	[7:0] bits 7-0 of PWM2's phase time	0x00
0x78d	PWM_PHASE2	R/W	[15:8] bits 15-8 of PWM2's phase time	0x00
0x78e	PWM_PHASE3	R/W	[7:0] bits 7-0 of PWM3's phase time	0x00
0x78f	PWM_PHASE3	R/W	[15:8] bits 15-8 of PWM3's phase time	0x00
0x790	PWM_PHASE4	R/W	[7:0] bits 7-0 of PWM4's phase time	0x00
0x791	PWM_PHASE4	R/W	[15:8] bits 15-8 of PWM4's phase time	0x00
0x792	PWM_PHASE5	R/W	[7:0] bits 7-0 of PWM5's phase time	0x00
0x793	PWM_PHASE5	R/W	[15:8] bits 15-8 of PWM5's phase time	0x00
0x794	PWM_TCMP0	R/W	[7:0] bits 7-0 of PWM0's high time or low time(if pola[0]=1)	0x00
0x795	PWM_TCMP0	R/W	[15:8] bits 15-8 of PWM0's high time or low time	0x00
0x796	PWM_TMAX0	R/W	[7:0] bits 7-0 of PWM0's cycle time	0x00
0x797	PWM_TMAX0	R/W	[15:8] bits 15-8 of PWM0's cycle time	0x00
0x798	PWM_TCMP1	R/W	[7:0] bits 7-0 of PWM1's high time or low time(if pola[1]=1)	0x00
0x799	PWM_TCMP1	R/W	[15:8] bits 15-8 of PWM1's high time or low	0x00

Address	Mnemonic	Type	Description	Reset Value
			time	
0x79a	PWM_TMAX1	R/W	[7:0] bits 7-0 of PWM1's cycle time	0x00
0x79b	PWM_TMAX1	R/W	[15:8] bits 15-8 of PWM1's cycle time	0x00
0x79c	PWM_TCMP2	R/W	[7:0] bits 7-0 of PWM2's high time or low time(if pola[2]=1)	0x00
0x79d	PWM_TCMP2	R/W	[15:8] bits 15-8 of PWM2's high time or low time	0x00
0x79e	PWM_TMAX2	R/W	[7:0] bits 7-0 of PWM2's cycle time	0x00
0x79f	PWM_TMAX2	R/W	[15:8] bits 15-8 of PWM2's cycle time	0x00
0x7a0	PWM_TCMP3	R/W	[7:0] bits 7-0 of PWM3's high time or low time(if pola[3]=1)	0x00
0x7a1	PWM_TCMP3	R/W	[15:8] bits 15-8 of PWM3's high time or low time	0x00
0x7a2	PWM_TMAX3	R/W	[7:0] bits 7-0 of PWM3's cycle time	0x00
0x7a3	PWM_TMAX3	R/W	[15:8] bits 15-8 of PWM3's cycle time	0x00
0x7a4	PWM_TCMP4	R/W	[7:0] bits 7-0 of PWM4's high time or low time(if pola[4]=1)	0x00
0x7a5	PWM_TCMP4	R/W	[15:8] bits 15-8 of PWM4's high time or low time	0x00
0x7a6	PWM_TMAX4	R/W	[7:0] bits 7-0 of PWM4's cycle time	0x00
0x7a7	PWM_TMAX4		[15:8] bits 15-8 of PWM4's cycle time	0x00
0x7a8	PWM_TCMP5	R/W	[7:0] bits 7-0 of PWM5's high time or low time(if pola[5]=1)	0x00
0x7a9	PWM_TCMP5	R/W	[15:8] bits 15-8 of PWM5's high time or low time	0x00
0x7aa	PWM_TMAX5	R/W	[7:0] bits 7-0 of PWM5's cycle time	0x00
0x7ab	PWM_TMAX5	R/W	[15:8] bits 15-8 of PWM5's cycle time	0x00
0x7ac	PWM_PNUM0	R/W	[7:0]PWM0 Pulse number in count mode and IR mode	0x00
0x7ad	PWM_PNUM0	R/W	[15:8]	0x00
0x7ae~ 0x7af	reserved			
0x7b0	PWM_MASK0	R/W	INT mask [0] PWM0 Pnumint 0: disable 1: Enable [1] rsvd [2] PWM0 frame int 0: disable 1: Enable [3] PWM1 frame int 0: disable 1: Enable [4] PWM2 frame int 0: disable 1: Enable [5] PWM3 frame int 0: disable 1: Enable [6] PWM4 frame int 0: disable 1: Enable	0x00

Address	Mnemonic	Type	Description	Reset Value
			[7] PWM5 frame int 0: disable 1: Enable	
0x7b1	PWM_INT0	R/W	INT status,write 1 to clear [0]:PWM0 pnumint(have sent PNUM pulses,PWM_NCNT==PWM_PNUM) [1]:rsvd [2]:PWM0 cycle done int(PWM_CNT==PWM_TMAX) [3]:PWM1 cycle done int(PWM_CNT==PWM_TMAX) [4]:PWM2 cycle done int(PWM_CNT==PWM_TMAX) [5]:PWM3 cycle done int(PWM_CNT==PWM_TMAX) [6]:PWM4 cycle done int(PWM_CNT==PWM_TMAX) [7]:PWM5 cycle done int(PWM_CNT==PWM_TMAX)	0x00
0x7b2	PWM_MASK1	R/W	[0]:PWM0 fifo mode stop int mask 0: disable 1: Enable [1]: PWM0 fifo mode fifocntint mask 0: disable 1: Enable	0x00
0x7b3	PWM_INT1	R/W	INT status,write 1 to clear [0]:fifo mode stop int,whenfifohassent the last waveform and becomes empty [1]: fifo mode cntint, when FIFO_NUM (0x7f4) is less than FIFO_NUM_LVL (0x7fc)	0x00
0x7b4	PWM_CNT0	R	[7:0]PWM0 cnt value	
0x7b5	PWM_CNT0		[15:8]PWM0 cnt value	
0x7b6	PWM_CNT1	R	[7:0]PWM1 cnt value	
0x7b7	PWM_CNT1		[15:8]PWM1 cnt value	
0x7b8	PWM_CNT2	R	[7:0]PWM2 cnt value	
0x7b9	PWM_CNT2		[15:8]PWM2 cnt value	
0x7ba	PWM_CNT3	R	[7:0]PWM3 cnt value	
0x7bb	PWM_CNT3		[15:8]PWM3 cnt value	
0x7bc	PWM_CNT4	R	[7:0]PWM4 cnt value	
0x7bd	PWM_CNT4		[15:8]PWM4 cnt value	
0x7be	PWM_CNT5	R	[7:0]PWM5 cnt value	
0x7bf	PWM_CNT5		[15:8]PWM5 cnt value	
0x7c0	PWM_NCNT0	R	[7:0]PWM0 pluse_cnt value	
0x7c1	PWM_NCNT0		[15:8]PWM0 pluse_cnt value	
0x7c2~ 0x7cb	reserved			
0x7cc	PWM_FM_CARRY_ EN0	R/W	[0]:waveform0_tail [1]:waveform0_head [2]:waveform1_tail	0x55

Address	Mnemonic	Type	Description	Reset Value
			[3]:waveform1_head [4]:waveform2_tail [5]:waveform2_head [6]:waveform3_tail [7]:waveform3_head 1b'1: carry enable 1b'0: send low level	
0x7cd	PWM_FM_CARRY_EN1	R/W	[0]:waveform4_tail [1]:waveform4_head [2]:waveform5_tail [3]:waveform5_head [4]:waveform6_tail [5]:waveform6_head [6]:waveform7_tail [7]:waveform7_head 1b'1: carry enable 1b'0: send low level	0x55
0x7ce~0x7cf	reserved			
0x7d0	WAVEFORM0_TAIL_CFG	R/W	[7:0] waveform0 tail carry pulse number	0x00
0x7d1	WAVEFORM0_TAIL_CFG	R/W	[15:8] waveform0 tail carry pulse number	0x00
0x7d2	WAVEFORM0_HEAD_CFG	R/W	[7:0] waveform0 head carry pulse number	0x00
0x7d3	WAVEFORM0_HEAD_CFG	R/W	[15:8] waveform0 head carry pulse number	0x00
0x7d4	WAVEFORM1_TAIL_CFG	R/W	[7:0] waveform1 tail carry pulse number	0x00
0x7d5	WAVEFORM1_TAIL_CFG	R/W	[15:8] waveform1 tail carry pulse number	0x00
0x7d6	WAVEFORM1_HEAD_CFG	R/W	[7:0] waveform1 head carry pulse number	0x00
0x7d7	WAVEFORM1_HEAD_CFG	R/W	[15:8] waveform1 head carry pulse number	0x00
0x7d8	WAVEFORM2_TAIL_CFG	R/W	[7:0] waveform2 tail carry pulse number	0x00
0x7d9	WAVEFORM2_TAIL_CFG	R/W	[15:8] waveform2 tail carry pulse number	0x00
0x7da	WAVEFORM2_HEAD_CFG	R/W	[7:0] waveform2 head carry pulse number	0x00
0x7db	WAVEFORM2_HEAD_CFG	R/W	[15:8] waveform2 head carry pulse number	0x00
0x7dc	WAVEFORM3_TAIL_CFG	R/W	[7:0] waveform3 tail carry pulse number	0x00
0x7dd	WAVEFORM3_TAIL_	R/W	[15:8] waveform3 tail carry pulse number	0x00

Address	Mnemonic	Type	Description	Reset Value
	CFG			
0x7de	WAVEFORM3_HEAD_CFG	R/W	[7:0] waveform3 head carry pulse number	0x00
0x7df	WAVEFORM3_HEAD_CFG	R/W	[15:8] waveform3 head carry pulse number	0x00
0x7e0	WAVEFORM4_TAIL_CFG	R/W	[7:0] waveform4 tail carry pulse number	0x00
0x7e1	WAVEFORM4_TAIL_CFG	R/W	[15:8] waveform4 tail carry pulse number	0x00
0x7e2	WAVEFORM4_HEAD_CFG	R/W	[7:0] waveform4 head carry pulse number	0x00
0x7e3	WAVEFORM4_HEAD_CFG	R/W	[15:8] waveform4 head carry pulse number	0x00
0x7e4	WAVEFORM5_TAIL_CFG	R/W	[7:0] waveform5 tail carry pulse number	0x00
0x7e5	WAVEFORM5_TAIL_CFG	R/W	[15:8] waveform5 tail carry pulse number	0x00
0x7e6	WAVEFORM5_HEAD_CFG	R/W	[7:0] waveform5 head carry pulse number	0x00
0x7e7	WAVEFORM5_HEAD_CFG	R/W	[15:8] waveform5 head carry pulse number	0x00
0x7e8	WAVEFORM6_TAIL_CFG	R/W	[7:0] waveform6 tail carry pulse number	0x00
0x7e9	WAVEFORM6_TAIL_CFG	R/W	[15:8] waveform6 tail carry pulse number	0x00
0x7ea	WAVEFORM6_HEAD_CFG	R/W	[7:0] waveform6 head carry pulse number	0x00
0x7eb	WAVEFORM6_HEAD_CFG	R/W	[15:8] waveform6 head carry pulse number	0x00
0x7ec	WAVEFORM7_TAIL_CFG	R/W	[7:0] waveform7 tail carry pulse number	0x00
0x7ed	WAVEFORM7_TAIL_CFG	R/W	[15:8] waveform7 tail carry pulse number	0x00
0x7ee	WAVEFORM7_HEAD_CFG	R/W	[7:0] waveform7 head carry pulse number	0x00
0x7ef	WAVEFORM7_HEAD_CFG	R/W	[15:8] waveform7 head carry pulse number	0x00
0x7f0	FIFO_INPUT	W	[3:0] write waveform number in fifo,first in [7:4] write waveform number in fifo,second in	0x00
0x7f1	FIFO_INPUT	W	[3:0] write waveform number in fifo,third in [7:4] write waveform number in fifo,forth in	0x00
0x7f2	reserved			
0x7f3	reserved			
0x7f4	FIFO_NUM	RO	fifo data number	
0x7f5	FIFO_WPTR	RO	fifo write pointer	

Address	Mnemonic	Type	Description	Reset Value
0x7f6	FIFO_RPTR	RO	fifo read pointer	
0x7f7	FIFO_SR	RO	fifo status [0]:1-fifo empty [1]:1-fifo full	
0x7f8	FIFO_CLEAR	W	Write 1 to clear fifo data	
0x7f9-0x7fb	reserved			
0x7fc	FIFO_NUM_LVL	R/W	FIFO numinttrigger level	0x00
0x7fd~0x7ff	reserved			

8.2 Enable PWM

Register PWM_EN (address 0x780)[5:0] serves to enable PWM5~PWM0 respectively via writing “1” for the corresponding bits.

8.3 Set PWM clock

PWM clock derives from system clock. Register PWM_CLK (address 0x781) serves to set the frequency dividing factor for PWM clock. Formula below applies:

$$F_{PWM} = F_{System\ clock} / (PWM_CLK+1)$$

8.4 PWM waveform, polarity and output inversion

Each PWM channel has independent counter and 3 status including “Delay”, “Count” and “Remaining”. Count and Remaining status form a signal frame.

8.4.1 PWM waveform

When PWM#n is enabled, PWM#n enters Delay status. By defaultPWM#n outputs Low level at Delay status. The Delay status duration, i.e. Phase time, is configured in register PWM_PHASE#n (address 0x788~0x793). Phase difference between PWM channels is allowed by different phase time configurations.

After Phase time expires, PWM#n exits Delay status and starts to send signal frames. First PWM#n is at Count status and outputs High level signal by default. When PWM#n counter reaches cycles set in register PWM_TCMP#n (address 0x794~0x795, 0x798~0x799, 0x79c~0x79d, 0x7a0~0x7a1, 0x7a4~0x7a5, 0x7a8~0x7a9), PWM#n enters Remaining status and outputs Low level till PWM#n cycle time configured in register PWM_TMAX#n (address 0x796~0x797, 0x79a~0x79b, 0x79e~0x79f, 0x7a2~0x7a3, 0x7a6~0x7a7, 0x7aa~0x7ab) expires.

An interruption will be generated at the end of each signal frame if enabled via register PWM_MASK (address 0x7b0[2:7]).

8.4.2 Invert PWM output

PWM#n and PWM#n_N output could be inverted independently via register PWM_CC0 (address 0x783) and PWM_CC1 (address 0x784). Whentheinversion bit is enabled, waveform of the corresponding PWM channel will be inverted completely.

8.4.3 Polarity for signal frame

By default, PWM#n outputs High level at Count status and Low level at Remaining status. When thecorresponding polarity bit is enabled via register PWM_CC2 (address 0x785[5:0]), PWM#n will output Low level at Count status and High level at Remaining status.

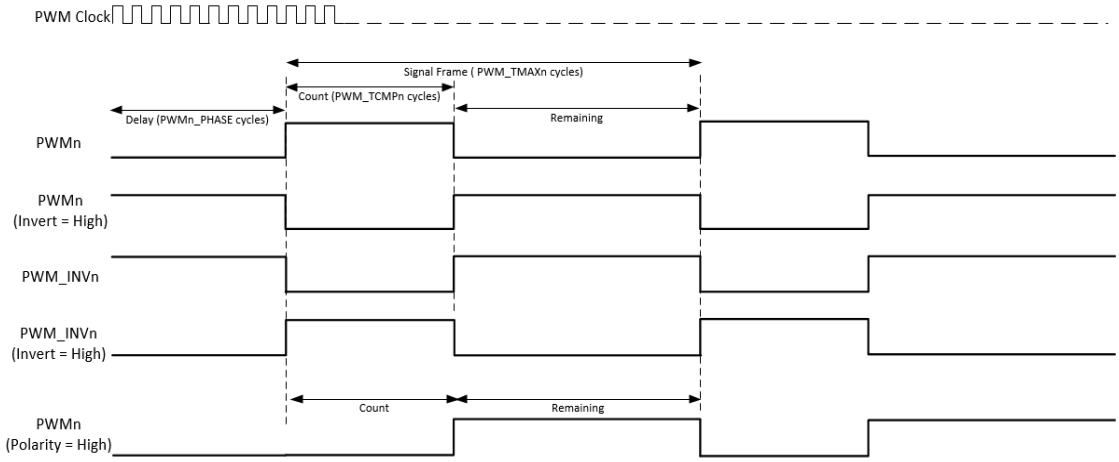


Figure 8- 1 PWM output waveform chart

8.5 PWM mode

8.5.1 Select PWM mode

PWM0 supports four modes,includingContinuous mode (normal mode, default), Counting mode, IR mode and IR FIFO mode.

PWM1~PWM5 only support Continuous mode.

Register PWM_MODE (address 0x782) serves to select PWM0 mode.

8.5.2 Continuous mode

PWM0~PWM5 all support Continuous mode. In this mode, PWM#n continuously sends out signal frames. PWM#n should be disabled via address 0x780 to stop it; when stopped, the PWM output will turn low immediately.

During Continuous mode, waveform could be changed freely. New configuration for PWM_TCMP#n and PWM_TMAX#n will take effect in the next signal frame.

After each signal frame is finished,corresponding PWM cycle done interrupt flag bit(0x7b1[2:7]) will be automatically set to 1b’1. If the interrupt is enabled by setting PWM_MASK0(address 0x7b0[2:7]) as 1b’1, a frame interruption will be generated. User needs to write 1b’1 to the flag bit to manually clear it.

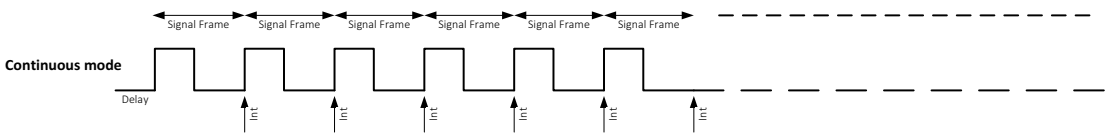


Figure 8-2 Continuous mode

8.5.3 Counting mode

Only PWM0 supports Counting mode. Address 0x782[1:0] should be set as 2b'01 to select PWM0 counting mode.

In this mode, PWM0 sends out specified number of signal frames which is defined as a pulse group. The number is configured via register PWM_PNUM0 (address 0x7ac~0x7ad).

After each signal frame is finished,PWM0 cycle done interrupt flag bit(0x7b1[2]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 (address 0x7b0[2]) as 1b'1, a frame interruption will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

After a pulse group is finished, PWM0 will be disabled automatically, and PWM0 pnuminterrupt flag bit(0x7b1[0]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK0 (address 0x7b0[0]) as 1b'1, a Pnuminterruptionwill be generated. User needs to write 1b'1 to the flag bit to manually clear it.

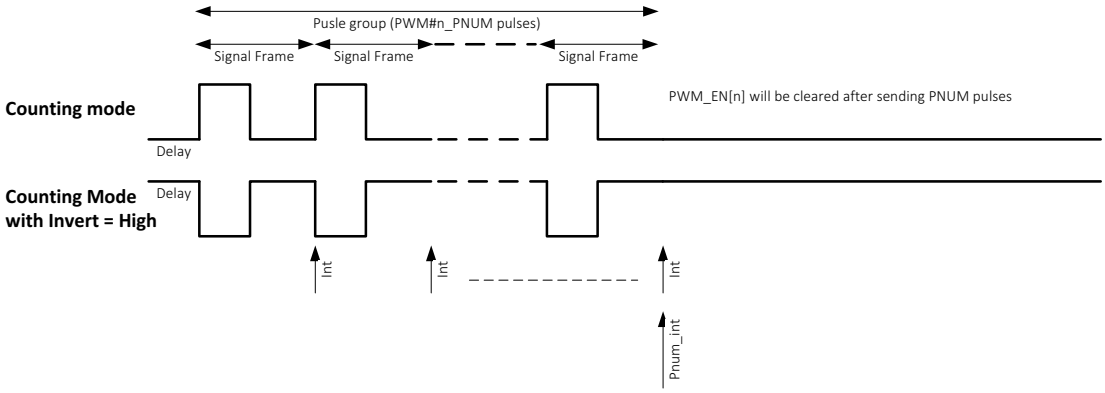


Figure 8-3 Counting mode (n=0)

Counting mode also serves to stop IR mode gracefully. Refer to **section 8.5.4** for details.

8.5.4 IR mode

Only PWM0 supports IR mode. Address 0x782[1:0] should be set as 2b'11 to select PWM0 IR mode.

In this mode, specified number of frames is defined as one pulse group. In contrast to Counting modewhere PWM0 stops after first pulse group is finished, PWM0will constantly send pulse groups in IR mode.

During IR mode, waveform could also be changed freely. New configuration for PWM_TCMP0, PWM_TMAX0and PWM_PNUM0will take effect in the next pulse group.

To stop IR mode and complete current pulse group, user can switch PWM0 from IR mode to Counting mode so that PWM0 will stop after current pulse group is finished. If PWM0 is disabled directly via PWM_EN (0x780[0]), PWM0 output will turn Low immediately despite of current pulse group.

After each signal frame/pulse group is finished,PWM0 cycle done interrupt flag bit(0x7b1[2])/PWM0 pnuminterrupt flag bit(0x7b1[0]) will be automatically set to 1b'1. A frame interruption/Pnum interruption will be generated (if enabledby setting address 0x7b0[2]/0x7b0[0] as 1b'1).

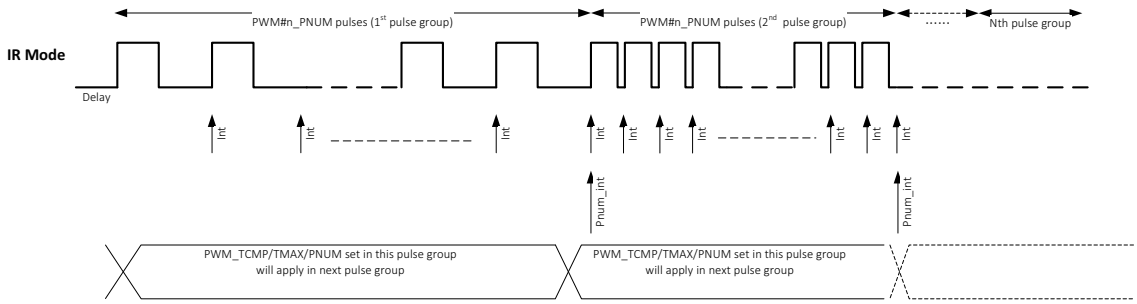


Figure 8-4 IR mode(n=0)

8.5.5 IR FIFO mode

IR FIFO mode is designed to allow IR transmission of long code patterns without the continued intervention of MCU, and it is designed as a selectable working mode on PWM0. The IR carrier frequency is divided down from the system clock and can be configured to be any normal IR frequencies, e.g. 36Khz, 38Khz, 40Khz, or 56Khz.

Only PWM0 supports IR FIFO mode.Address 0x782[4,1:0] should be set as 3b'111 to select PWM0 IR FIFO mode.

In this mode, an IR packet can be disassembled to multiple waveforms consisting of preamble, data 0, data 1, repeat andend. Each waveform is considered to have the following structure with two repetitive patterns: the first part is designated as HEAD and the second part is designated as TAIL.

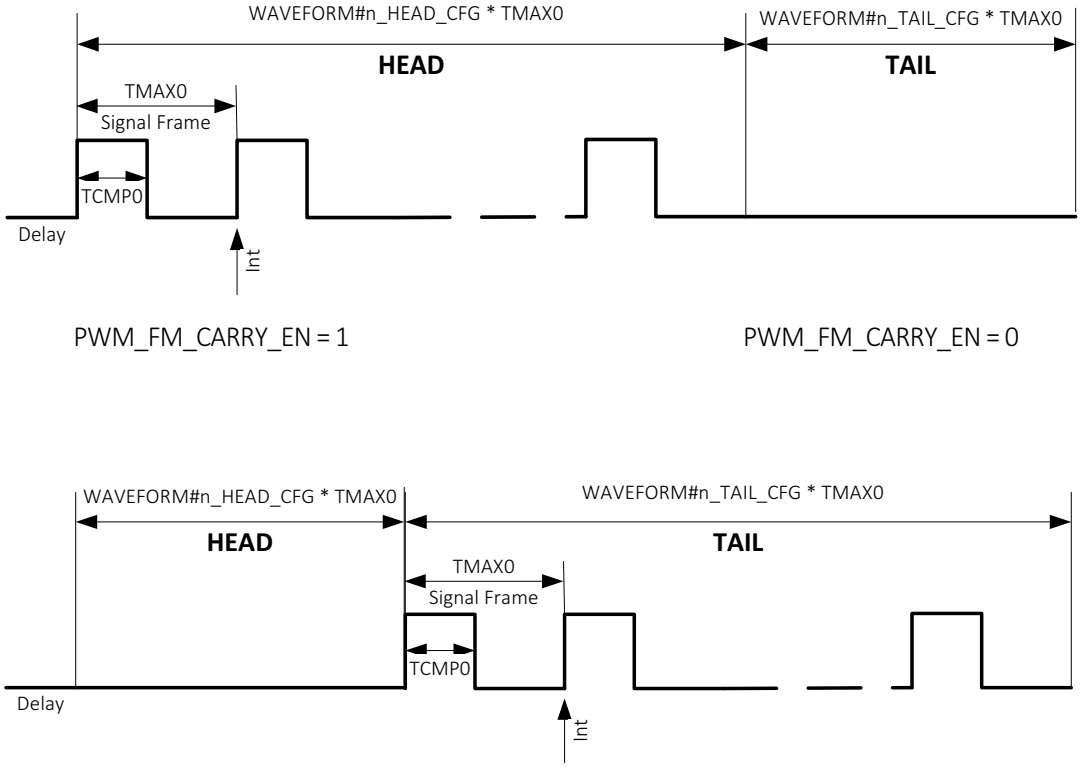


Figure 8- 5 Waveform structure in IR FIFO mode

Up to eight waveform definitions are supported and each is configurable via three registers:

- PWM_FM_CARRY_EN: a 2byte register, each bit used to enable/disable the carrier for the HEAD and TAIL part respectively;
- WAVEFORM_HEAD_CFG: configure the length of the HEAD part as number of cycles of the nominal carrier frequency;
- WAVEFORM_TAIL_CFG: configure the length of the TAIL part as number of cycles of the nominal carrier frequency.

Each waveform is identified with a unique indexnumberranging from 0 to 7. Once the waveforms are clearly defined, they can be used in a FIFO to define the IR code.

The FIFO is 70 in depth, i.e. 35 bytes with four bits for each waveform index and thus up to 70 data (waveform index numbers) can be stored in the FIFO at any time. Waveforms should be added in sequential order by writing corresponding index number into FIFO via a two-byte FIFO_INPUT register (addresses 0x7f0~0x7f1). Up to four index numbers can be written every time.

After PWM0 isenabledvia PWM_EN (address 0x780[0]), the configured waveforms will be output from PWM0 in sequence. As long as FIFO doesn't overflow, user can continue to add waveforms during IR waveforms sending process, and long IR code that exceeds the FIFO depth can be implemented this way. After all waveforms are sent, FIFO becomes empty, PWM0 will be

disabled automatically.

Example:

Suppose carrier frequency is 38K, system clock frequency is 32M, duty cycle is 1/3,andtheformat of IR code to be sent is shown as below:

1) Preamble waveform: 9ms carrier + 4.5ms low level.
2) Data 1 waveform: 0.56ms carrier + 0.56ms low level.
3) Data 0 waveform: 0.56ms carrier + 1.69ms low level.
4) Repeat waveform: 9ms carrier + 2.25ms low level + 0.56ms carrier. Repeat waveform duration is 11.81ms, interval between two adjacent repeat waveforms is 108ms.
5) End waveform: 0.56ms carrier.

User can follow the steps below to configure related registers:

Step1: Set carrier frequency as 38K.

Set **PWM_TMAX0** as 0x34a (i.e. 32M/38K=842=0x34a).

Since duty cycle is 1/3, set**PWM_TCMP0** as 0x119 (i.e. 842/3=0x119).

Step2: Select suitable waveform types. For example, set preamble waveform as waveform0, set data 1 waveform as waveform1, set data 0 waveform as waveform2, set 9ms carrier and 2.25ms low level of repeat waveform as waveform3, set 0.56ms carrier as waveform4, set waveform between two adjacent repeat waveforms (i.e. 108ms-11.81ms low level) as waveform5.

Thus a repeat waveform consists of a waveform3 and a waveform4, and end waveform consists of a waveform4.

Step3: Configure each waveform.

Waveform0 configuration:

(1) Since head is 9ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[1] to 1b'1 to enable carrier, and set **WAVEFORM0_HEAD_CFG**(addresses 0x7d3~0x7d2) as0x156(i.e. 9ms/(1000/38000 ms)= 342).

(2) Since tail is 4.5ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[0] to 1b'0 to send low level,and set **WAVEFORM0_TAIL_CFG** (addresses 0x7d1~0x7d0) as0xab (i.e. 4.5ms/(1000/38000 ms)=171).

Waveform1 configuration:

(1) Since head is 0.56ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[3] to 1b'1 to enable carrier, and set **WAVEFORM1_HEAD_CFG**(addresses 0x7d7~0x7d6) as 0x15 (i.e. 0.56ms/(1000/38000 ms)=21).

(2) Since tail is 0.56ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[2] to 1b'0 to send low level, and set **WAVEFORM1_TAIL_CFG** (addresses 0x7d5~0x7d4) as 0x15 (i.e. 0.56ms/(1000/38000 ms)=21).

Waveform2 configuration:

(1) Since head is 0.56ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[5] to 1b'1 to enable carrier, and set **WAVEFORM2_HEAD_CFG**(addresses 0x7db~0x7da) as 0x15 (i.e. 0.56ms/(1000/38000 ms)=21).

(2) Since tail is 1.69ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[4] to 1b'0 to send low level, and set **WAVEFORM2_TAIL_CFG** (addresses 0x7d9~0x7d8) as 0x40 (i.e. 1.69ms/(1000/38000 ms)=64).

Waveform3 configuration:

(1) Since head is 9ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[7] to 1b'1 to enable carrier, and set **WAVEFORM3_HEAD_CFG**(addresses 0x7df~0x7de) as 0x156 (i.e. 9ms/(1000/38000 ms)=342).

(2) Since tail is 2.25ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cc) bit[6] to 1b'0 to send low level, and set **WAVEFORM3_TAIL_CFG** (addresses 0x7dd~0x7dc) as 0x56 (i.e. 2.25ms/(1000/38000 ms)=86).

Waveform4 configuration:

Divide 0.56ms carrier into 0.26ms carrier (head) and 0.3ms carrier (tail).

(1) Since head is 0.26ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cd) bit[1] to 1b'1 to enable carrier, and set **WAVEFORM4_HEAD_CFG**(addresses 0x7e3~0x7e2) as 0x0a (i.e. 0.26ms/(1000/38000 ms)=10).

(2) Since tail is 0.3ms carrier, set **PWM_FM_CARRY_EN0** (address 0x7cd) bit[0] to 1b'1 to enable carrier, and set **WAVEFORM4_TAIL_CFG** (addresses 0x7e1~0x7e0) as 0x0b (i.e. 0.3ms/(1000/38000 ms)=11).

Waveform5 configuration:

Divide (108ms-11.81ms) low level into 48.1ms low level (head) and 48.1ms low level (tail).

(1) Since head is 48.1ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cd) bit[3] to 1b'0 to send low level, and set **WAVEFORM5_HEAD_CFG**(addresses 0x7e7~0x7e6) as 0x724 (i.e. 48.1ms/(1000/38000 ms)=1828).

(2) Since tail is 48.1ms low level, set **PWM_FM_CARRY_EN0** (address 0x7cd) bit[2] to 1b'0 to send low level, and set **WAVEFORM5_TAIL_CFG** (addresses 0x7e5~0x7e4) as 0x724 (i.e. 48.1ms/(1000/38000 ms)=1828).

Step4: Write waveform index numbers into FIFO via FIFO_INPUT register.

For each validwriting, 1~4 numbers are allowed, and the two bytes of FIFO_INPUT register (addresses 0x7f1 and 0x7f0) **must** be configured at the same time. To add a number, write FIFO_INPUT with "0xfffn" (n indicates waveform index number); to add two numbers, write FIFO_INPUT with "0xffnn"; to add three numbers, write FIFO_INPUT with "0xfnnn"; to add four numbers, write FIFO_INPUT with "0xnnnn".

Step5: Enable PWM0 via setting PWM_EN (address 0x780) bit[0] to 1b'1. Then corresponding waveforms with index number in FIFO will be sent out from PWM0 pin. As long as FIFO is not full, user can continue to add waveforms during IR waveforms sending process.

User can read the register FIFO_NUM (address 0x7f4) to view current remaining number of data (waveform) in FIFO to be sent.

The register FIFO_SR (address 0x7f7) serves to check FIFO status.

When the FIFO_NUM value is less than the FIFO_NUM_LVL (address 0x7fc), the fifo count interrupt flag bit (address 0x7b3[1]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK1 (address 0x7b2[1]) as 1b'1, a FIFO mode count interrupt will be generated. User needs to write 1b'1 to the flag bit to manually clear it. In implementation, this can be used as a trigger for user to write additional waveform index into the FIFO.

After all waveforms are sent, FIFO becomes empty, PWM0 will be disabled automatically (address 0x780[0] is cleared). The FIFO mode stop interrupt flag bit (address 0x7b3[0]) will be automatically set to 1b'1. If the interrupt is enabled by setting PWM_MASK1 (address 0x7b2[0]) as 1b'1, a FIFO mode stop interrupt will be generated. User needs to write 1b'1 to the flag bit to manually clear it.

The FIFO_CLEAR register (address 0x7f8) serves to clear data in FIFO. Writing 1b'1 to this register will clear all data in the FIFO. Note that the FIFO can only be cleared when not in active transmission.

8.6 PWM interrupt

There are 9interrupt sources from PWM function.

After each signal frame, PWM#n(n=0~5)will generate a frame-done IRQ (Interrupt Request) signal.

In Counting mode and IR mode, PWM0 will generate a Pnum IRQ signal after completing a pulse group.

In IR FIFO mode, PWM0 will generatea FIFO mode count IRQ signal when the FIFO_NUM value is less than the FIFO_NUM_LVL,and will generate a FIFO mode stop IRQ signal after FIFO becomes empty.

Interrupt status can be cleared via register PWM_INT (address 0x7b1[7:2,0]) and PWM_INT1 (address 0x7b3[1:0]).

9 Audio

9.1 Audio input path

There are four types of audio input path: digital microphone (DMIC), Codec (I2S), USB and analog input channel (AMIC), which is selectable by writing address 0xb11[3:2].

Address 0xb11[4] should be set as 1b'1/1b'0 to select mono/stereo input for audio input processing module.

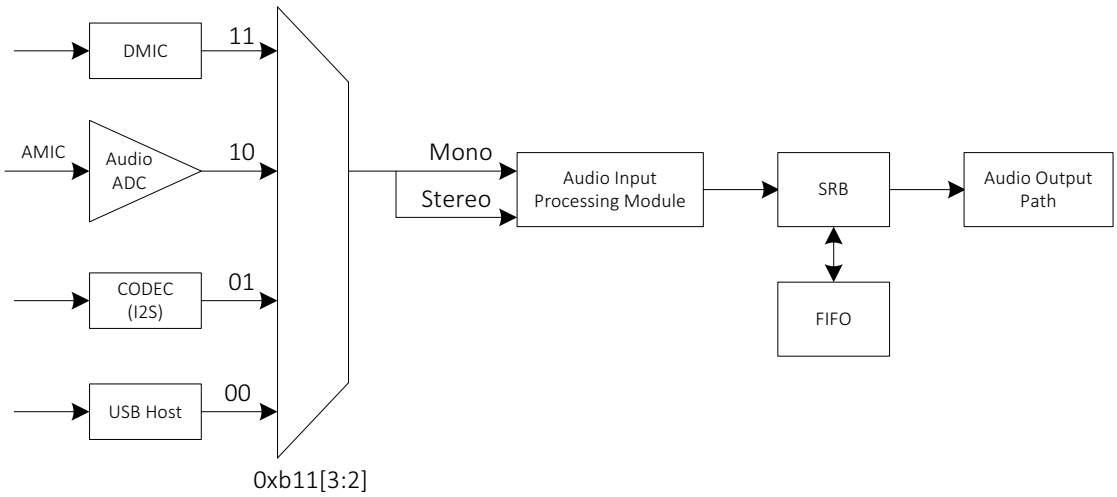


Figure 9- 1 Audio input path

Table 9- 1 Audio data flow direction

Data Path		Target SRAM		
		FIFO0	FIFO1	FIFO2
DMIC	Decimation filter /LPF/Down Sample/ HPF/ALC	√	√	×
I2S		√	√	×
USB		√	√	×
ANALOG CH0		√	√	×
ANALOG CH1		√	√	×
ANALOG CH2		×	×	√

9.1.1 AMIC input

Address 0xb11[3:2] should be set as 2b'10 to select AMIC as audio input.

A programmable stereo PGA (Programmable Gain Amplifier) with adjustable gain is built in for AMIC. AMIC input channel can carry out signal amplification via the PGA.

After implementing AD conversion for selected AMIC input signal, data of 3 analog channels (CH0~CH2) will be generated. Data of CH0/CH1 are sent to the audio input processing module, while data of CH2 are directly written into FIFO.

9.1.2 DMIC input

Stereo digital microphone (DMIC) interface is also supported in the ST17H58. Address 0xb11[3:2] should be set as 2b'11 to select DMIC as audio input.

DMIC interface includes one configurable clock line and one data line. Address 0xb11[1:0] serves to set the rising/falling edge of clock signal at which to sample data of DMIC, and it should not be set as 2b'11. Address 0xb11[7:6] serves to enable/mask the DMIC channel which samples data at rising/falling edge. Generally, address 0xb11 bit[1:0] is set as 2b'01 and bit[7:6] is set as 2b'01 to enable DMIC0 which samples data at rising edge of clock.

After data sampling of DMIC interface, sign extension and audio input processing, the signal can be written into FIFO.

9.1.3 I2S input

Address 0xb11[3:2] should be set as 2b'01 to select I2S as audio input.

Digital I2S audio interface supports Master mode only, 16-bit data width, and variable sampling rate: 8K/16K/22.05K/24K/32K/44.1K/48K. The sampling rate is determined by I2S clock.

For I2S clock configuration, please refer to section **4.3.3 I2S clock**.

Address 0x560[5]/[4]/[1] should be set to "1" to enable I2S interface, I2S Recorder and I2S Player, respectively. I2S interface includes one configurable clock line, one data line and one channel selection line. Data generated by the audio codec will be written into FIFO after implementing conversion via I2S Recorder and audio input processing.

9.1.4 USB Host input

Address 0xb11[3:2] should be set as 2b'00 to select USB as audio input.

Packet transmitted by USB Host will be written into FIFO after implementing conversion via USB Interface and audio input processing.

Address 0x560[3] should be set to 1b'1 to enable ISO player.

9.1.5 DFIFO

As shown in Table 9-1, for any type of audio input path, the data will be finally written into DFIFO (DMA FIFO) 0, 1 or 2.

Address 0xb10[0]/[1]/[2] should be set as 1b'1 to enable audio input of DFIFO 0~2.

DFIFO supports auto mode and manual mode. It's highly recommended to clear address 0xb2c[0] to select auto mode.

Take DFIFO0 as an example:

- ✧ Address 0xb00, 0xb01 and 0xb03 serve to set base address for DFIFO0, i.e. starting address to write/read data into/from DFIFO0.
- ✧ Address 0xb02 serves to set depth (i.e. the maximum data number) for DFIFO0. Suppose address 0xb02 is set as 0x01, then the DFIFO0 depth is 4 words, i.e. 16 bytes.
- ✧ Current data number (difference value of write-pointer and read-pointer) in DFIFO0 can be

read from address 0xb20 and 0xb21.

- ✧ User can check current DFIFO0 read pointer/write pointer location by reading address 0xb14~0xb15/0xb16~0xb17.
- ✧ When current data number in DFIFO0 is less than the underflow threshold set in address 0xb0c, address 0xb13 bit[0] and bit[4] will be set as 1b'1 successively, and a FIFO0 low interrupt will be generated if enabled via 0xb10[4].

  Address 0xb13[0] will be automatically cleared when the data number in DFIFO0 is no less than the threshold; address 0xb13[4] needs to be cleared manually.
- ✧ When current data number in DFIFO0 is more than the overflow threshold set in address 0xb0d, address 0xb13 bit[1] and bit[5] will be set as 1b'1 successively, and a FIFO0 high interrupt will be generated if enabled via 0xb10[5].

  Address 0xb13[1] will be automatically cleared when the data number in DFIFO0 is no more than the threshold; address 0xb13[5] needs to be cleared manually.

9.2 Audio input processing

Audio input processing mainly includes configurable decimation filter, LPF (Low Pass Filter), Down-sample module, HPF (High Pass Filter) and ALC (Automatic Level Control). The decimation filter, LPF, Down-sample module, HPF and ALC can be enabled or bypassed via address 0xb11[5] and 0xb40 [6]/[7]/[4]/[5].

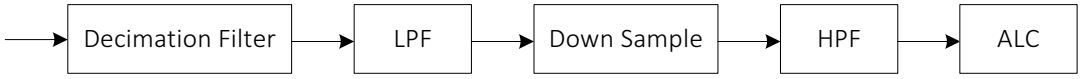


Figure 9- 2 Audio input processing

9.2.1 Decimation filter

Address 0xb11[5] should be cleared to enable decimation filter.

The decimation filter serves to down-sample the mono or stereo input (e.g. DMIC) data to required audio data playback rate (e.g. 48K or 32K).

Down-sampling rate is configurable by writing address 0xb12[3:0]/0xb12[2:0] and 0xb35[2]: If 0xb35[2] is set as 1b'1, the down-sampling rate can be configured as 1~16 via 0xb12[3:0]; if 0xb35[2] is set as 1b'0, the down-sampling rate can be configured as 16, 32, 64, 128 or 256 via 0xb12[2:0].

Address 0xb12[7:4] serves to adjust decimation filter output by right shift, so that the data after down-sampling won't exceed data bit width.

9.2.2 LPF

Address 0xb40[6] should be cleared to enable the LPF. The LPF serves to conduct frequency compensation.

9.2.3 Down-sample

If the Down-sample module is enabled by setting address 0xb40[7] to 1b'1, it will down-sample the data from LPF with fixed ratio of 2.

9.2.4 HPF

Address 0xb40[4] should be cleared to enable the HPF.

The HPF serves to eliminate internal DC offset to ensure audio amplification range.

The HPF output is adjustable via setting the parameter in address 0xb40[3:0].

9.2.5 ALC

The ALC supports analog mode and digital mode, and it mainly serves to regulate input volume level automatically or manually in each mode.

The analog mode is designed only for AMIC input, while the digital mode applies to all audio input types.

In analog mode, input volume level is regulated via PGA; while in digital mode, input volume level is regulated via Multiplier/Divider.

9.2.5.1 Auto regulate in analog mode

In this case, address 0xb54[0] should be set as 1b'1, and analog AGC (Automatical Gain Control) parameters can be adjusted via address 0xb41~0xb54[7:1] and 0xb55~0xb5d.

Initial value of the PGA gain can be set via address 0xb5d.

The result of (AMIC input* current PGA gain) is compared with high volume target (ALC_VOL_THH), low volume target (ALC_VOL_THL) and volume noise level (ALC_VOL_THN); meanwhile, it will be compared with a VAD noise threshold (ALC_VAD_THN) to judge noise signal and help to regulate the PGA gain. The PGA gain will be automatically adjusted according to the comparison results, and it should be within the range from minimum PGA gain (ALC_VOL_L/ALC_VOL_R) to maximum PGA gain (ALC_VOL_H).

- ✧ Address 0xb45[0] and 0xb44[7:3] serve to set integer part of high volume target in unit of dB; while address 0xb44[2:0] serve to set fractional part of high volume target in unit of dB.
- ✧ Address 0xb47[0] and 0xb46[7:3] serve to set integer part of low volume target in unit of dB; while address 0xb46[2:0] serve to set fractional part of low volume target in unit of dB.
- ✧ Address 0xb49[0] and 0xb48[7:3] serve to set integer part of noise level in unit of dB; while address 0xb48[2:0] serve to set fractional part of noise level in unit of dB.
- ✧ Address 0xb4b[0] and 0xb4a[7:3]serve to set integer part of VAD noise threshold in unit of dB; while address 0xb4a[2:0] serve to set fractional part of VAD noise threshold in unit of dB.
- ✧ Address 0xb43[6:0] (ALC_VOL_H) serves to set the maximum PGA gain, while 0xb41[6:0]/0xb42[6:0] (ALC_VOL_L/ALC_VOL_R) serves to set the minimum PGA gain in left/right channel.
- ✧ User can check current PGA gain in left/right channel by reading address 0xb5e/0xb5f.

9.2.5.2 Manual regulate in analog mode

In this case, PGA gain can be adjusted by either of the following two manual modes.

- ✧ Manual mode 1: Address 0xb63[7] should be set as 1b'1 to select manual mode 1. In this mode, the PGA consists of two stages of amplifiers including pre-amplifier and post-amplifier, and each stage has configurable gain. Address 0xb63[6] serves to set gain for the pre-amplifier (Boost-stage) as 26dB or 46dB; while address 0xb63[5:0] serves to set gain for the post-amplifier (Gain-stage) as -10dB (0x0)~14dB (0x30) with step of 0.5dB.
- ✧ Manual mode 2: Address 0xb63[7] and 0xb54[0] should be cleared to select manual mode 2. In this mode, address 0xb61[6:0]/0xb65[6:0] serves to set target gain value for left/right channel, while 0xb60 serves to set the speed for PGA gain to reach the target gain value. Current PGA gain can be read from address 0xb62[6:0]/0xb66[6:0] which changes until the target gain value is reached. When PGA gain reaches the target value, address 0xb61[7]/0xb65[7] will be set as 1b'1.

9.2.5.3 Auto regulate in digital mode

In this case, address 0xb54[0] should be set as 1b'0, and 0xb41[7]/0xb42[7] should be set as 1b'1 to enable auto regulation mode for left/right channel.

ALC (Automatical Level Control) parameters can be adjusted via address 0xb41~0xb55.

Initial value of the coefficient is 0x20 (equivalent to 1). The valid range of coefficient is 0~0x3f: if the coefficient is within 0~0x20, the divider is used; if the coefficient is within 0x20~0x3f, the multiplier is used.

The result of (Input volume* current coefficient) is compared with high threshold (ALC_VOL_THH0), low threshold (ALC_VOL_THL0) and noise threshold (ALC_VOL_THN0). The coefficient will be automatically adjusted according to the comparison result, and it should be within the range from minimum value (ALC_VOL_L/ALC_VOL_R) to maximum value (ALC_VOL_H).

- ✧ Address 0xb44[6:0] serve to set high threshold.
- ✧ Address 0xb46[6:0] serve to set low threshold.
- ✧ Address 0xb48[6:0] serve to set noise threshold.
- ✧ Address 0xb43[5:0] (ALC_VOL_H) serves to set the maximum coefficient, while 0xb41[5:0]/0xb42[5:0] (ALC_VOL_L/ALC_VOL_R) serves to set the minimum coefficient in left/right channel.
- ✧ User can check current volume in left/right channel by reading address 0xb4d/0xb4e.

9.2.5.4 Manual regulate in digital mode

In this case, address 0xb54[0] should be set as 1b'0, and 0xb41[7]/0xb42[7] should be set as 1b'0 to enable manual regulation mode for left/right channel.

The coefficient of multiplier/divider is configurable via the register ALC_VOL_L (address 0xb41[5:0] / ALC_VOL_R (address 0xb42[5:0]) for left/right channel.

9.2.6 Register configuration

Table 9- 2 Register configuration related to audio input processing

Address	Mnemonic	Type	Description	Reset value
0x67	I2S step	R/W	[7] I2S clock enable [6:0] i2s step	33
0x68	I2S Mod	R/W	I2s mod I2s clock = fhs*step[6:0]/mod[7:0], Mod need be larger than or equal to 2*step. Fhs refer 0x70 FHS_sel	2
0xb00	DFIFO0_BADR0	RW	FIFO0 base address [7:0]	0x00
0xb01	DFIFO0 BADR1	RW	FIFO0 base address [15:8]	0x40
0xb02	DFIFO0_DEPTH	RW	FIFO depth=FIFO0_DEPTH*4words	0x7f
0xb03	DFIFO0_BADR2	RW	FIFO0 base address [18:16]	0x04
0xb04	DFIFO1_BADR0	RW	FIFO1 base address [7:0]	0x00
0xb05	DFIFO1 BADR1	RW	FIFO1 base address [15:8]	0x48
0xb06	DFIFO1_DEPTH	RW	FIFO depth=FIFO1_DEPTH*4words	0x7f
0xb07	DFIFO1_BADR2	RW	FIFO1 base address [18:16]	0x04
0xb08	DFIFO2_BADR0	RW	FIFO2 base address [7:0]	0x00
0xb09	DFIFO2 BADR1	RW	FIFO2 base address [15:8]	0x3c
0xb0a	DFIFO2_DEPTH	RW	FIFO depth=FIFO2_DEPTH*4words	0x3f
0xb0b	DFIFO2_BADR2	RW	FIFO2 base address [18:16]	0x04
0xb0c	DFIFO0_LLEV	RW	FIFO0 low level	0x20
0xb0d	DFIFO0_HLEV	RW	FIFO0 high level	0x60
0xb0e	DFIFO1_HLEV	RW	FIFO1 high level	0x20
0xb0f	DFIFO2_HLEV	RW	FIFO2 high level	0x20
0xb10	DFIFOMODE	RW	DFIFO enable [0]: enable audio input of FIFO0 [1]: enable audio input of FIFO1 [2]: enable audio input of FIFO2 [3]: enable audio output of FIFO0 [4]: FIFO0 low interrupt enable [5]: FIFO0 high interrupt enable [6]: FIFO1 high interrupt enable [7]: FIFO2 high interrupt enable	F9
0xb11	DFIFOAIN	RW	[0]: 1: D-MIC0 use data at rising edge of clock 0: at falling edge [1]: 1: D-MIC1 use data at rising edge of clock 0: at falling edge [3:2]: audio input select 0: USB; 1: I2S; 2: ADC; 3: D-MIC	21

Address	Mnemonic	Type	Description	Reset value
			[4]: mono (1)/stereo (0) input [5]: enable (0) or bypass (1) decimation filter input [6]:dmic_raise_chn_not_en [7]:dmic_fall_chn_not_en (generally disabled)	
0xb12	DFIFODEC	RW	[3:0]: Decimation ratio when cic_mode set as 1 0~15: [3:0] + 1 [2:0]: Decimation ratio when cic_mode set as 0 0: 16; 1: 32; 2: 64; 3: 128: else: 256 See 0xb35[2] for cic_mode [7:4]: cic bit shift[3:0]	5b
0xb13	DFIFOIRQST	R	[0]: FIFO0 low interrupt flag. Auto clear. [1]: FIFO0 high interrupt flag. Auto clear. [2]: FIFO1 high interrupt flag. Auto clear. [3]: FIFO2 high interrupt flag. Auto clear. [4]: FIFO0 low. Need manual clear. [5]: FIFO0 high. Need manual clear. [6]: FIFO1 high. Need manual clear. [7]: FIFO1 high. Need manual clear.	00
0xb14	DFIFO0_RPTRL	RO	FIFO READ PTR low byte	00
0xb15	DFIFO0_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00
0xb16	DFIFO0_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb17	DFIFO0_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb18	DFIFO1_RPTRL	RO	FIFO READ PTR low byte	00
0xb19	DFIFO1_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00
0xb1a	DFIFO1_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb1b	DFIFO1_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb1c	DFIFO2_RPTRL	RO	FIFO READ PTR low byte	00
0xb1d	DFIFO2_RPTRH	RO	[3:0]:FIFO READ PTR high byte	00
0xb1e	DFIFO2_WPTRL	RO	FIFO WRITE PTR low byte	00
0xb1f	DFIFO2_WPTRH	RO	[3:0]:FIFO WRITE PTR high byte	00
0xb20	DFIFO0_NUML	RO	FIFO DATA NUMBER low byte	00
0xb21	DFIFO0_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb22	reserved			
0xb23	reserved			
0xb24	DFIFO1_NUML	RO	FIFO DATA NUMBER low byte	00
0xb25	DFIFO1_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb26	reserved			
0xb27	reserved			
0xb28	DFIFO2_NUML	RO	FIFO DATA NUMBER low byte	00
0xb29	DFIFO2_NUMH	RO	FIFO DATA NUMBER high byte	00
0xb2a	reserved			
0xb2b	reserved			

Address	Mnemonic	Type	Description	Reset value
0xb2c	DFIFO_MANUAL	RW	[0]: 0: fifo auto mode 1: enable fifo manual mode	00
0xb2d	reserved			
0xb2e	reserved			
0xb2f	reserved			
0xb30	DFIFO_MAN_DAT0	W	Fifo manual mode data in[7:0]	00
0xb31	DFIFO_MAN_DAT1	W	Fifo manual mode data in[15:8]	00
0xb32	DFIFO_MAN_DAT2	W	Fifo manual mode data in[23:16]	00
0xb33	DFIFO_MAN_DAT3	W	Fifo manual mode data in[31:24]	00
0xb34	ALC_SFT	RW	Before alc data shift	00
0xb35	AUDIO_CFG	RW	[0]:adclnr valid select [1]: adc 64/63 option [2]: cic_mode	00
0xb36	ADC_MUL	RW	Adccalmul	00
0xb37	ADC_BIAS	RW	Adccal bias	00
0xb38-0xb3f	reserved			
0xb40	ALC_HPF_LPF_BYPASS	RW	[3:0]: Parameter to adjust HPF output. [4]: bypass HPF 1: bypass HPF, 0: use HPF [5]: bypass ALC 1: bypass ALC, 0: use ALC [6]: bypass LPF 1: bypass LPF, 0: use LPF [7]: enable double down sample 1: enable, 0: disable	03
0xb41	ALC_VOL_L	RW	ALC left channel setting [5:0]: minimum volume in digital mode [6:0]:minimumpga gain in analog mode [7]: select auto or manual mode for digital mode 1: auto mode, 0: manual mode	20
0xb42	ALC_VOL_R	RW	ALC right channel setting [5:0]: minimum volume in digital mode [6:0]:minimumpga gain in analog mode [7]: select auto or manual mode for digital mode 1: auto mode, 0: manual mode	20
0xb43	ALC_VOL_H	RW	[5:0]: maximum volume in digital mode [6:0]:maximum pga gain in analog mode	33
0xb44	ALC_VOL_THH0	RW	[6:0]: high threshold in digital mode volume decrease condition vmax> ALC_VOL_THH [2:0]: high volume target dbfra in analog mode [7:3]:high volume target dbint[4:0] in	60

Address	Mnemonic	Type	Description	Reset value
			analog mode	
0xb45	ALC_VOL_THH1	RW	[0]:high volume target dbint[5] in analog mode	00
0xb46	ALC_VOL_THL0	RW	[6:0]: low threshold in digital mode volume increase condition vmax< ALC_VOL_THL [2:0]: low volume target dbfra in analog mode [7:3]:low volume target dbint[4:0] in analog mode	40
0xb47	ALC_VOL_THL1	RW	[0]:low volume target dbint[5] in analog mode	00
0xb48	ALC_VOL_THN0	RW	[6:0]: noise threshold in digital mode [2:0]: volume noise level dbfra in analog mode [7:3]:volume noise level dbint[4:0] in analog mode	02
0xb49	ALC_VOL_THN1	RW	[0]:volume noise level dbint[5] in analog mode	00
0xb4a	ALC_VAD_THN0	RW	[2:0]: VAD noise level dbfra in analog mode [7:3]:VAD noise level dbint[4:0] in analog mode	02
0xb4b	ALC_VAD_THN1	RW	[0]:VAD noise level dbint[5] in analog mode	00
0xb4c	ALC_VOL_STEP	RW	[3:0]: increase step [7:4]: decrease step	11
0xb4d	ALC_VOL_L	R	Vmax in digital manual mode vol in digital auto mode	
0xb4e	ALC_VOL_R	R	Vmax in digital manual mode vol in digital auto mode	
0xb4f	reserved			
0xb50	ALC_PEAK_TICK_L	RW	[7:0]:low byte of tick	00
0xb51	ALC_PEAK_TICK_H	RW	[7:0]:high byte of tick	02
0xb52	ALC_DEC_TICK	RW	Decrease tick	00
0xb53	ALC_NOI_TICK	RW	Noise decrease tick	ff
0xb54	ALC_CFG	RW	[0]:analog mode en [1]:noise en [2]:vad en [3]:vadiir en [4]:ana_iir_en [5]:iir tick clear en [6]:vad max en [7]:peak noise tick same en	02
0xb55	ALC_COEF_IIR	RW	Iircoef	fa
0xb56	ALC_DAT_MASK	RW	In analog mode when gain changes,	00

Address	Mnemonic	Type	Description	Reset value
			configure number of data be masked	
0xb57	ALC_INC_SPD	RW	In analog mode, pga gain increase speed	0a
0xb58	ALC_INC_MAX	RW	In analog mode, pga gain increase max in a peak tick cycle	03
0xb59	ALC_DEC_SPD	RW	In analog mode, pga gain decrease speed	08
0xb5a	ALC_DEC_MAX	RW	In analog mode, pga gain decrease max in a peak tick cycle	06
0xb5b	ALC_NOI_SPD	RW	In analog mode, pga gain decrease speed when as noise	06
0xb5c	ALC_NOI_MAX	RW	In analog mode, pga gain decrease max in a peak tick cycle when as noise	06
0xb5d	PGA_GAIN_INIT	RW	Pga gain initial value	00
0xb5e	PGA_GAIN_L	R	Pga gain in auto mode	00
0xb5f	PGA_GAIN_R	R	Pga gain in auto mode	00
0xb60	PGA_MAN_SPEED	RW	Pga manual mode judge speed	00
0xb61	PGA_MAN_TARGET_L	RW	Left channel [6:0]: pga manual mode target [7]: pga judge done	
0xb62	PGA_VALUE_L	R	Left channel [6:0]: PGA value in manual mode [7]: Mute status (R). Enable (0)/Mute (1) PGA output (W).	
0xb63	PGA_FIX_VALUE	RW	[6:0]: Pga gain fix value [7]: pga gain fix enable	
0xb64	PGA_R_L	RW	[0]: change PGA L R CHANNEL	
0xb65	PGA_MAN_TARGET_R	RW	Right channel [6:0]: pga manual mode target [7]: pga judge done	
0xb66	PGA_VALUE_R	R	Right channel [6:0]: PGA value in manual mode [7]: Mute status (R). Enable (0)/Mute (1) PGA output (W).	

9.3 Audio output path

Audio output path mainly includes Rate Matching module, SDMDAC (Sigma-Delta Modulation DAC) and I2S Player. The audio data fetched from SRAM is processed by the Rate Matching module, then transferred to the SDM/I2S Player as the input signal.

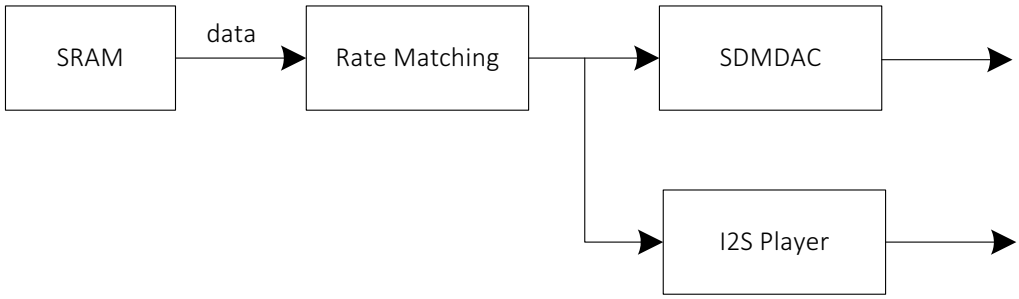


Figure 9- 3 Audio output path

9.3.1 Rate Matching

The rate matching block performs clock rate conversion and data synchronization between two domains: the input audio data is fetched from SRAM which works in system clock domain with 24Mhz/32Mhz/48Mhz clocks and the SDM/I2S which works between 4Mhz and 8Mhz.

When needed, the audio data from SRAM is interpolated to the SDM/I2S input rate. If the audio sampling rate is ClkUsbIn (e.g. 48Khz), and the working clock of SDM/I2S is aclk_i, then the interpolation ratio is given as follows:

$$\frac{ClkUsbIn}{aclk_i} = \frac{step_i}{0x8000}$$

Where step_i[19:0] is configured in addresses 0x567~0x565.

Linear interpolation or delay interpolation is used as shown below.

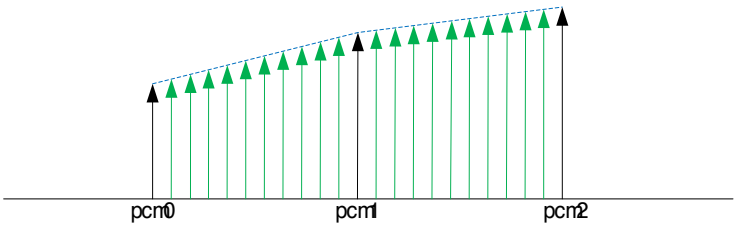


Figure 9- 4 Linear interpolation

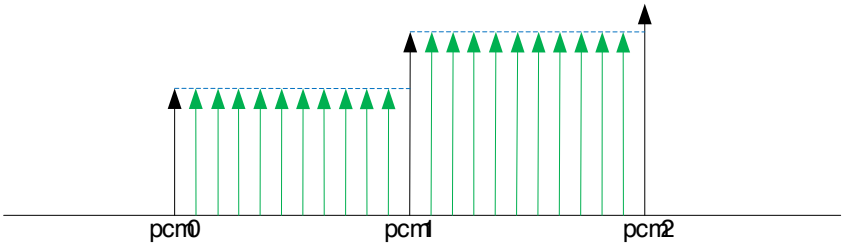


Figure 9- 5 Delay interpolation

9.3.2 SDM

The SDM takes 16bits audio data from SRAM and provides 1bit modulated output. Only a simple passive filter network is needed to drive audio device directly.

Dither control can be added to the SDM to avoid spurs in output data. There are three dithering options: PN sequence, PN sequence with Shapping, and DC constant; only one type of input is allowed any time.

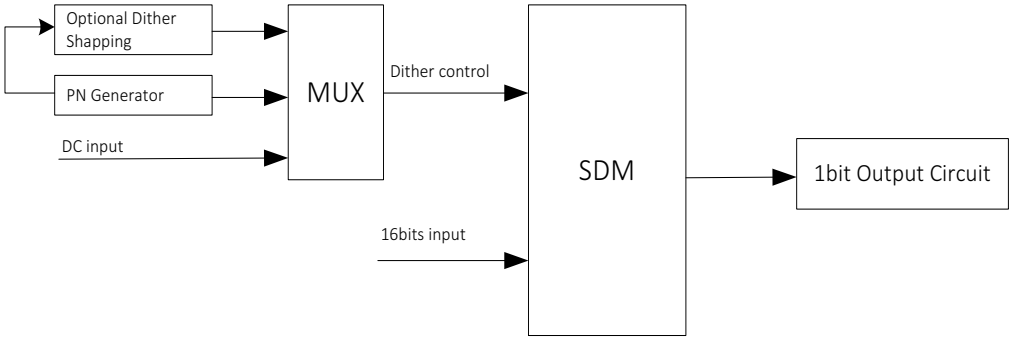


Figure 9- 6 Block diagram of SDM

9.3.3 Register configuration

Address 0x560[4:1] should be set to “1” to enable I2S recorder/ISO player/SDM player/I2S player, while bit[0] is to select either mono or stereo audio output. Address 0x560[7] should be set to “1” to enable the HPF in audio output path.

Register VOL_CTRL (address 0x562) serves to adjust volume level.

Address 0x563[2] serves to select either linear interpolation or delay interpolation for the rate matching block: Setting bit[2] to “1” is to select linear interpolation, while clearing the bit is to select delay interpolation.

Input for SDM Dither control is selectable via addresses 0x56b[6:5]), 0x563[6:5] and 0x568~0x569.

For the **left channel**:

1. Address 0x56b[5] should be set to 1b’1 to select constant DC input. When DC input is used, addresses 0x56c~0x56d serve to configure the input constant value.
2. Address 0x56b[5] should be set to 1b’0 to use PN generator. Address 0x563[5] serves to enable/mask dither shapping module. There are two PN generators to generate random dithering sequence; address 0x568 bit[6]/bit[5] is enabling bit of the two PN generators.
   1) To select PN sequence as input, address 0x56b[5] and 0x563[5] should be set to 0, 0x568[6]/[5]/[6:5] should be set to 1.
   2) To select PN sequence with Shapping as input, address 0x56b[5] should be set to 0, 0x563[5] and 0x568[6]/[5]/[6:5] should be set to 1.

   When PN sequence or PN with Shapping is used, address 0x568[4:0]/0x569[4:0] determines the number of bits (ranging from 0 to 16) used in PN1/PN2 generator.

For the **right channel**:

1. Address 0x56b[6] should be set to 1b'1 to select constant DC input. When DC input is enabled, addresses 0x56e~0x56f serve to configure the input constant value.
2. Address 0x56b[6] should be set to 1b'0 to use PN generator. Address 0x563[6] serves to enable/mask dither shapping module. There are two PN generators to generate random dithering sequence; address 0x569 bit[6]/bit[5] is enabling bit of the two PN generators.
   1) To select PN sequence as input, address 0x56b[6] and 0x563[6] should be set to 0, 0x569[6]/[5]/[6:5] should be set to 1.
   2) To select PN sequence with Shapping as input, address 0x56b[6] should be set to 0, 0x563[6] and 0x569[6]/[5]/[6:5] should be set to 1.

   When PN sequence or PN with Shapping is used, address 0x56a[4:0]/0x56b[4:0] determines the number of bits (ranging from 0 to 16) used in PN1/PN2 generator.

Address 0x567, 0x566 and 0x565[7:4] are to set step_i[19:0] for the rate matching block, while address 0x564 is to tune the step_i value. The step_i should be in accordance with the aclk_i provided by SDM/I2S clock.

Table 9- 3 Register configuration related to audio output path

Address	Mnemonic	Type	Description	Reset value
0x560	AUDIO_CTRL	RW	[0]1--mono mode audio output, 0--stereo mode audio output [1]1--enable I2S player, 0--disable I2S player [2]1--enable SDM player, 0--disable SDM player [3]1--enable ISO player, 0--disable ISO player [4]1--enable I2S recorder , 0--disable I2s recorder [5]1--enable interface of I2S, 0--disable interface of I2S [6]1--enable GRP,0--disable GRP [7]1--enable HPF,0--disable HPF	04
0x561	MIDGRP	RW	[7:0]-- Middle of GRP	40
0x562	VOL_CTRL	RW	[0]--Add a quarter [1]--Add a half [6:2]--shift left [7]1--mute, 0--normal	40
0x563	PWM_CTRL	RW	[0]1--not multiply 2 when PWM, 0--mutiply2 [1]1--PWM, 0--not PWM [2]1-linear interpolate, 0-delay interpolate [4:3] reserved [5]1-left Shappingused, 0-left	64

Address	Mnemonic	Type	Description	Reset value
			Shapping not used [6]1-right Shapping used, 0-right Shapping not used	
0x564	ASCL_TUNE	RW	[7:0] tune step_i for rate matching block	01
0x565	I2S_FCT	RW	[3:0]factor to generate I2S clock [7:4]low 4 bits of rate matching blockstep_i[3:0]	90
0x566	ASCL_STEP0	RW	[7:0] middle byte of rate matching blockstep_i[11:4]	c4
0x567	ASCL_STEP1	RW	[7:0]high byte of rate matching blockstep_i[19:12]	00
0x568	PN1_LEFT	RW	[4:0]bits used in pn1 of left channel, range from 0 to 16 [5]1-pn2 of left enable, 0-pn2 of left disable [6]1-pn1 of left enable, 0-pn1 of left disable	50
0x569	PN2_LEFT	RW	[4:0]bits used in pn2 of left channel, range from 0 to 16 [5]1-pn2 of right enable, 0-pn2 of right disable [6]1-pn1 of right enable, 0-pn1 of right disable	40
0x56a	PN1_RIGHT	RW	[4:0]bits used in pn1 of right channel, range from 0 to 16 [5]1-clk2a is the same as audio clk, 0-clk2a is negated [6]1-exchange data in between SDMs, 0-not exchanged	10
0x56b	PN2_RIGHT	RW	[4:0]bits used in pn2 of right channel, range from 0 to 16 [5]1-left channel use constvalue, 0-left channel use pn [6]1-right channel use const value, 0-right channel use pn	00
0x56c	CONST_LEFT0	RW	[7:0]low byte of left channel const, i.e, const_l[7:0]	00
0x56d	CONST_LEFT1	RW	[7:0]high byte of left channel const, i.e.const_l[15:8]	00
0x56e	CONST_RIGHT0	RW	[7:0]low byte of right channel const, i.econst_r[7:0]	00
0x56f	CONST_RIGHT1	RW	[7:0]high byte of right channel const, i.econst_r[15:8]	00

10 Quadrature Decoder

The ST57H25F512embeds one quadrature decoder (QDEC) which is designed mainly for applications such as wheel. The QDEC implements debounce function to filter out jitter on the two phase inputs, and generates smooth square waves for the two phase.

10.1 Input pin selection

The QDEC supports two phase input; each input is selectable from the 8 pins of PortD, PortC, PortBand PortAvia setting address 0xd2[2:0] (for channel a)/0xd3[2:0] (for channel b).

Table 10- 1 Input pin selection

Address 0xd2[2:0]/0xd3[2:0]	Pin
0	ANA_A<2>
1	ANA_A<3>
2	ANA_B<6>
3	ANA_B<7>
4	ANA_C<2>
5	ANA_C<3>
6	ANA_D<6>
7	ANA_D<7>

Note: To use corresponding IO as QDEC input pin, it's needed first to enable GPIO function, enable "IE" (1) and disable "OEN" (1) for this IO.

10.2 Common mode and double accuracy mode

The QDEC embeds an internalhardwarecounter, which is not connected with bus.

Address 0xd7[0]serves to select common mode or double accuracy mode.

For each wheel rolling step, two pulse edges (rising edge or falling edge) are generated.

If address 0xd7[0] is cleared to select common mode, the QDEC Counter value (real time counting value) is increased/decreased by 1 only when the same rising/fallingedgesare detected from the two phase signals.

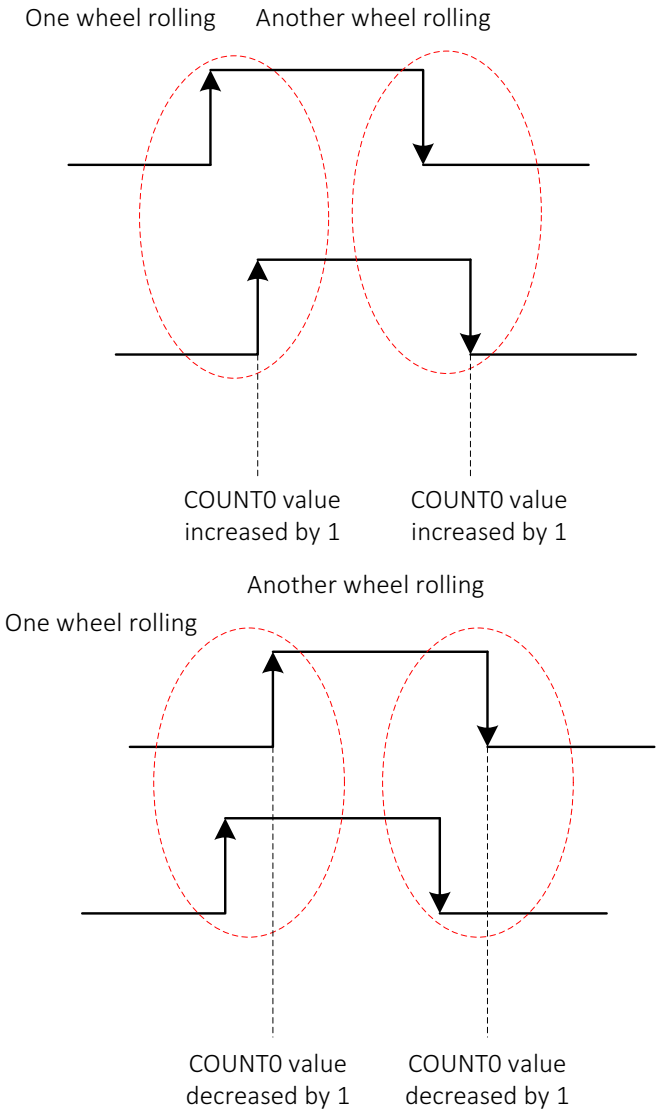


Figure 10- 1 Common mode

If address 0xd7[0] is set to 1b'1 to select double accuracy mode, the QDEC Counter value (real time counting value) is increased/decreased by 1 on each rising/falling edge of the two phase signals; the COUNT0 will be increased/decreased by 2 for one wheel rolling.

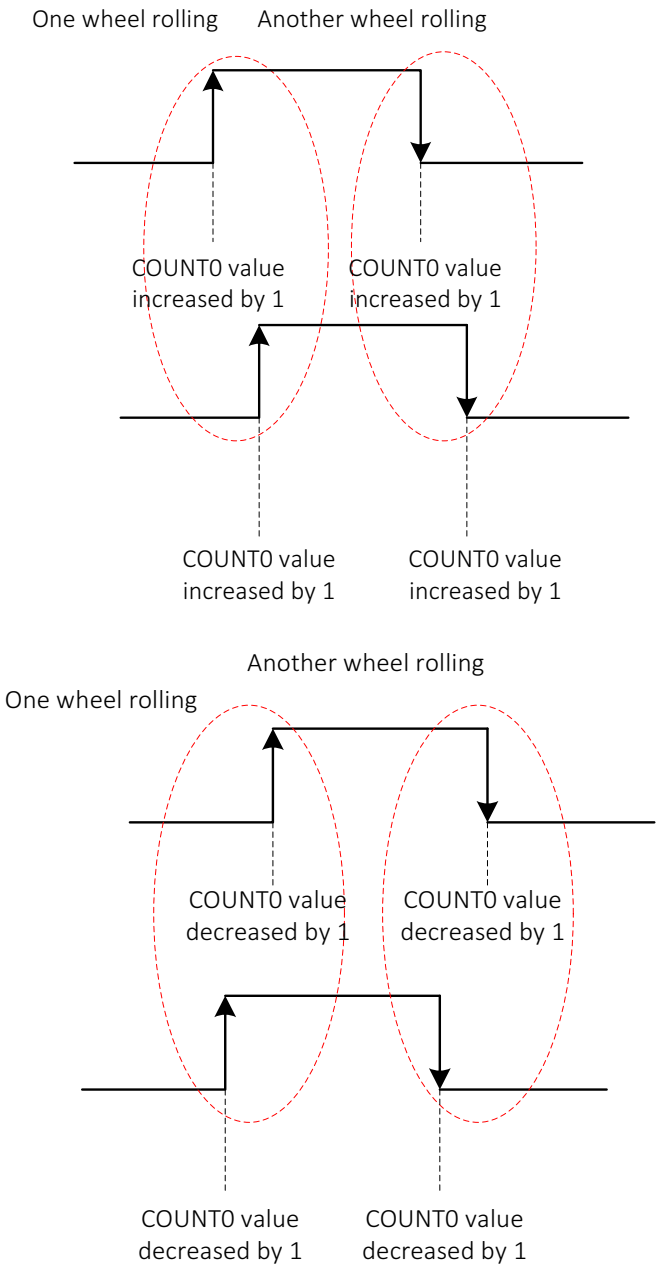


Figure 10- 2 Double accuracy mode

10.3 Read real time counting value

Neither can Hardware Counter value be read directly via software, nor can the counting value in address 0xd0 be updated automatically.

To read real time counting value, first write address 0xd8[0] with 1b'1 to load Hardware Counter data into the QDEC_COUNT register, then read address 0xd0.

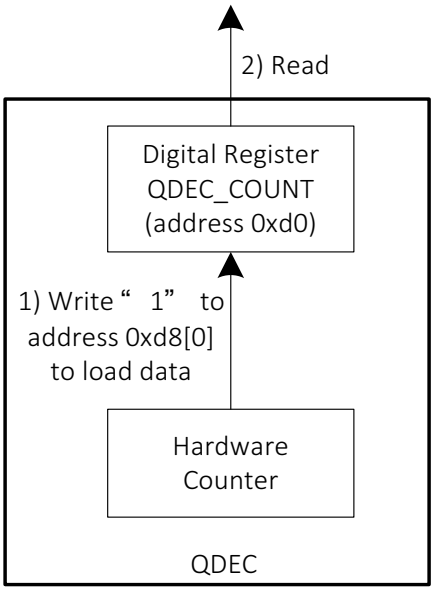


Figure 10- 3 Read real time counting value

10.4 QDEC interrupt

Address 0xd4[0] serves to enable or mask QDEC interrupt.

If address 0xd4[0] is set to 1b'1 to enable QDEC interrupt, whenever counter value changes, an QDEC IRQ is asserted and address 0xd5[0] is set to 1b'1 automatically. Writing 1b'1 to address 0xd5[0] can clear the interrupt flag bit.

10.5 QDEC reset

Address0xd6[0] serves to reset the QDEC. The QDEC Counter value is cleared to zero.

10.6 Other configuration

The QDEC supports hardware debouncing. Address 0xd1[2:0] serves to set filtering window duration. All jitter with period less than the value will be filtered out and thus does not trigger count change.

Address 0xd1[4] serves to set input signal initial polarity.

Address 0xd1[5] serves to enable shuttle mode. Shuttle mode allows non-overlapping two phase signals as shown in the following figure.

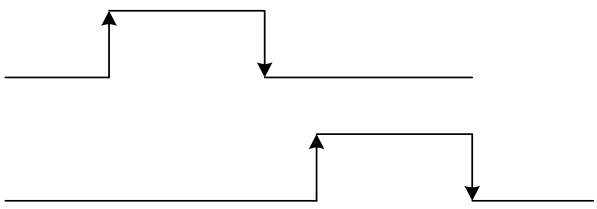


Figure 10- 4 Shuttle mode

10.7 Register table

Table 10- 2 Register table for QDEC

Address	Mnemonic	Type	Description	Reset value
0xd0	QDEC_COUNT	R	QDEC Counting value (read to clear): Pulse edge number	
0xd1	QDEC_CC	R/W	[2:0] : filter time (can filter 2^n *clk_32k*2 width deglitch) [4]: pola, input signal pola 0: no signal is low, 1: no signal is high [5]:shuttle mode 1 to enable shuttle mode	
0xd2	QDEC_CHNA	R/W	[2:0] QDEC input pin select for channel a choose 1 of 8 pins for input channel a 7~0: {pd[7:6],pc[3:2],pb[7:6],pa[3:2]}	0x00
0xd3	QDEC_CHNB	R/W	[2:0] QDEC input pin select for channel b choose 1 of 8 pins for input channel b 7~0: {pd[7:6],pc[3:2],pb[7:6],pa[3:2]}	0x01
0xd4	QDEC_MASK	R/W	[0]Interrupt mask. 1: enable, 0: mask	0x00
0xd5	QDEC_INT	R/W	[0]Interrupt flag. Write 1 to clear.	
0xd6	QDEC_RST	R/W	[0]Write 1 to reset QDEC	0x0
0xd7	QDEC_DOUBLE	R/W	[0]Enable double accuracy mode	0x0
0xd8	DATA_LOAD	R/W	[0]write 1 to load data when load completes it will be 0	

11 SAR ADC

The ST17H58 integrates one SAR ADC module, which can be used to sampleanalog input signals such as battery voltage, temperature sensor, mono or stereo audio signals, as well as internal noise signal.

11.1 Power on/down

The SAR ADC is disabled by default. To power on the ADC, the analog register adc_pd (afe_1P2V_reg124<5>) should be set as 1b'0.

11.2 ADC clock

ADC clock is derived from external 24MHz crystal source, with frequency dividing factor configurable via the analog register adc_clk_div (afe_1P2V_reg116<2:0>).

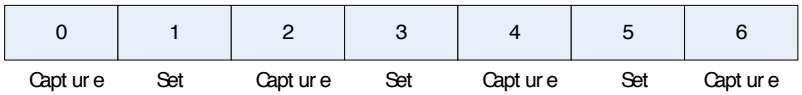
ADC clock frequency = 24M/(adc_clk_div+1)

11.3 ADC control in auto mode

11.3.1 Set max state and enable channel

The SAR ADC supports up to four channels including RNG channel, left channel, right channel and Misc channel. The RNG channel only contains one "Capture" state (state 0), while the left, right and Misc channels all consist of one "Set" state and one "Capture" state.

- The analog register r_max_scnt (afe_1P2V_reg114<6:4>) serves to set the max state index. As shown in the example below, the r_max_scnt should be set as 0x06.



- The RNG/left/Misc channel can be enabled independently via r_en_rns (afe_1P2V_reg114<3>), r_en_left (afe_1P2V_reg114<0>), r_en_misc (afe_1P2V_reg114<2>).
- Only when the left channel is enabled, can the right channel be enabled via r_en_right (afe_1P2V_reg114<1>).
- To sample internal noise signals, the RNG channel should be enabled. To sample mono audio signals, the left channel should be enabled. To sample stereo audio signals, both the left channel and the right channel should be enabled.

11.3.2 "Set" state

The length of "Set" state for left, right and Miscchannelis configurable via the analog register r_max_s (afe_1P2V_reg113<3:0>).

"Set" state duration = r_max_s * 24M clock period.

Each "Set" state serves to set ADC control signals for current channel via corresponding analog registers, including:

- ✧ adc_en_diff: afe_1P2V_reg108<4> (left channel), afe_1P2V_reg108<5> (right channel), afe_1P2V_reg108<6> (Misc channel). Select single-end or differential input mode.

- ✧ adc_ain_p: afe_1P2V_reg104<7:4> (Misc channel), afe_1P2V_reg105<7:4> (left channel), afe_1P2V_reg106<7:4> (right channel). Select single-end input in single-end mode, or select positive input in differential mode.

- ✧ adc_ain_n: afe_1P2V_reg104<3:0> (Misc channel), afe_1P2V_reg105<3:0> (left channel), afe_1P2V_reg106<3:0> (right channel). Select negative input in differential mode.

- ✧ adc_vref: afe_1P2V_reg103<1:0> (left channel), afe_1P2V_reg103<3:2> (right channel), afe_1P2V_reg103<5:4> (Misc channel). Set reference voltage V_{REF}. ADC maximum input range is the determined by the ADC reference voltage.

- ✧ adc_sel_ai_scale: afe_1P2V_reg122<7:6>. Set scaling factor for ADC analog input as 1 (default), 1/2, 1/4 or 1/8.

  By setting this scaling factor, ADC maximum input range can be extended based on the V_{REF}.

  For example, suppose the V_{REF} is set as 1.2V:

  Since the scaling factor is 1 by default, the ADC maximum input range should be 0~1.2V (single mode)/-1.2V~+1.2V (differential mode).

  If the scaling factor is set as 1/2, ADC maximum input range should change to 0~2.4V (single mode)/-2.4V~+2.4V (differential mode).

- ✧ adc_res: afe_1P2V_reg107<1:0> (left channel), afe_1P2V_reg107<5:4> (right channel), afe_1P2V_reg108<1:0> (Misc channel). Set resolution as 8/10/12/14 bits.

  ADC data is always 15-bitformat no matter what the resolution is set.For example, 14 bits resolution indicates ADC data consists of 14-bit valid data and 1-bit sign extension bit.

- ✧ adc_tsamp: afe_1P2V_reg109<3:0> (left channel), afe_1P2V_reg109<7:4> (right channel), afe_1P2V_reg110<3:0> (Misc channel). Set sampling time which determines the speed to stabilize input signals.

  Sampling time = adc_tsamp * ADC clock period.

  The lower sampling cycle, the shorter ADC convert time.

- ✧ pga_boost, pga_gain: Set PGA gain in Boost stage and Gain stage. See PGA section.

11.3.3 “Capture” state

For the RNG channel, the “Capture” state serves to capture random number sequence (RNS) derived from internal noise.

For the left, right and Misc channels, at the beginning of each “Capture” state, run signal is issued automatically to start an ADC sampling and conversion process; at the end of each “Capture” state, ADC output data is captured.

✧ The length of “Capture” state for RNS and Misc channel is configurable via the analog register r_max_mc[9:0] (afe_1P2V_reg113<7:6>, afe_1P2V_reg111<7:0>).

“Capture” state duration for RNS &Misc channel= r_max_mc * 24M clock period.

✧ The length of “Capture” state for left and right channel is configurable via the analog register r_max_c[9:0] (afe_1P2V_reg113<5:4>, afe_1P2V_reg112<7:0>).

“Capture” state duration for left & right channel = r_max_c * 24M clock period.

✧ The RNG channel will output 1bit random number, and user can read the analog register rng[15:0] (afe_1P2V_reg118<7:0>~ afe_1P2V_reg117<7:0>) to obtain the 16-bit random number consisting of 1-bit real-time random number output and 15-bit buffered RNS.

✧ The “VLD” bit (afe_1P2V_reg120<7>) will be set as 1b’1 at the end of “Capture” state to indicate the ADC data is valid, and this flag bit will be cleared automatically.

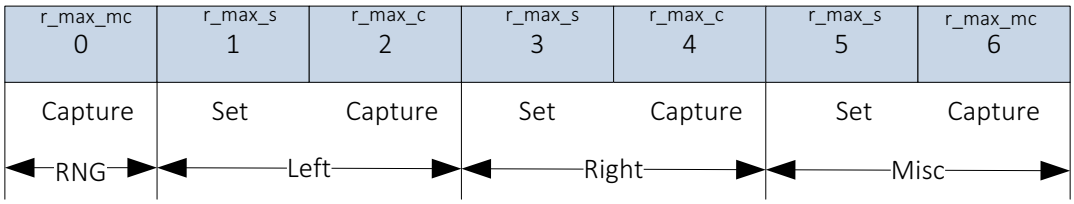
✧ The 15-bit ADC output data for Misc channel can be read from the analog register adc_dat[14:0] (afe_1P2V_reg120<6:0>, afe_1P2V_reg119<7:0>).

11.3.4 Usage cases

11.3.4.1 Case 1: 4-channel sampling for random number, stereo audio and Misc

In this case, the RNG, left, right and Misc channels should be enabledby setting afe_1P2V_reg114<3:0> as 0xf, and the max state index should be set as “6” by setting afe_1P2V_reg114<6:4> as 0x6.

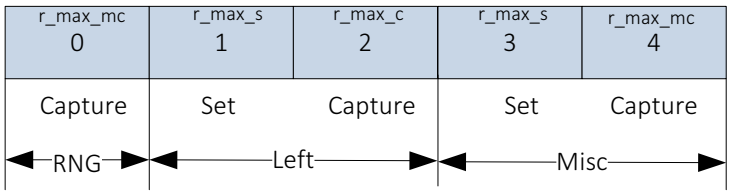
The total duration is (2*r_max_mc+3*r_max_s+2*r_max_c) * 24M clock period.



11.3.4.2 Case 2: 3-channel sampling for random number, mono audio and Misc

In this case, afe_1P2V_reg114<3:0> should be set as 0xd, so as to enable the RNG, left and Misc channels and disable the right channel, the max state index should be set as “4” by setting afe_1P2V_reg114<6:4> as 0x4.

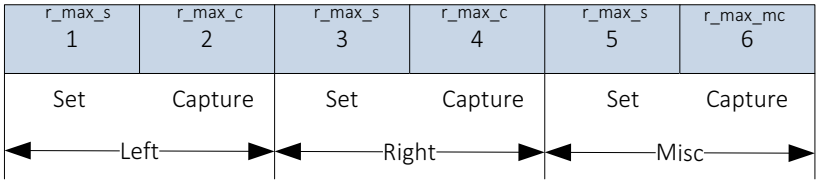
The total duration is (2*r_max_mc+2*r_max_s+1*r_max_c) * 24M clock period.



11.3.4.3 Case 3: 3-channel sampling for stereo audio and Misc

In this case, afe_1P2V_reg114<3:0> should be set as 0x7, so as to enable the left, right and Misc channels and disable the RNG channel, the max state index should be set as “6” by setting afe_1P2V_reg114<6:4> as 0x6.

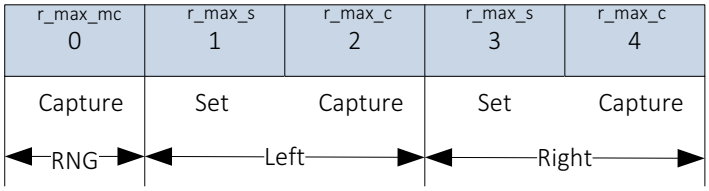
The total duration is (1*r_max_mc+3*r_max_s+2*r_max_c) * 24M clock period.



11.3.4.4 Case 4: 3-channel sampling for random number and stereo audio

In this case, afe_1P2V_reg114<3:0> should be set as 0xB, so as to enable the RNG, left and right channels and disable theMisc channel, the max state index should be set as “4” by setting afe_1P2V_reg114<6:4> as 0x4.

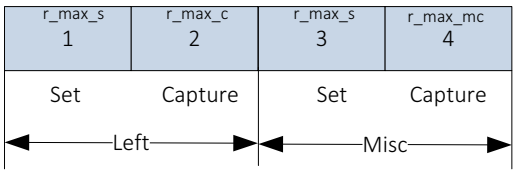
The total duration is (1*r_max_mc+2*r_max_s+2*r_max_c) * 24M clock period.



11.3.4.5 Case 5: 2-channel sampling for mono audio and Misc

In this case, afe_1P2V_reg114<3:0> should be set as 0x5, so as to enable theleft and Miscchannels and disable theRNGand right channels, the max state index should be set as “4” by setting afe_1P2V_reg114<6:4> as 0x4.

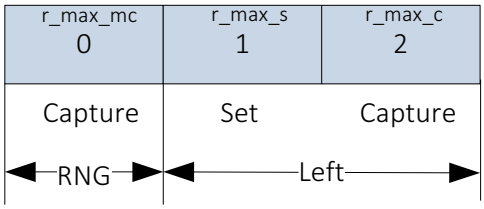
The total duration is (1*r_max_mc+2*r_max_s+1*r_max_c) * 24M clock period.



11.3.4.6 Case 6: 2-channel sampling for random number and mono audio

In this case, afe_1P2V_reg114<3:0> should be set as 0x9, so as to enable the RNG and left channels and disable theright and Misc channels, the max state index should be set as “2” by setting afe_1P2V_reg114<6:4> as 0x2.

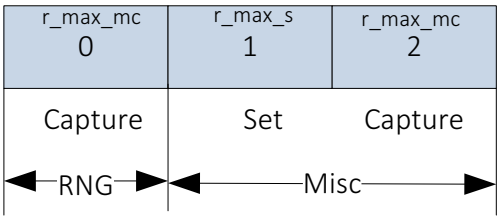
The total duration is (1*r_max_mc+1*r_max_s+1*r_max_c) * 24M clock period.



11.3.4.7 Case 7: 2-channel sampling for random number and Misc

In this case, afe_1P2V_reg114<3:0> should be set as 0xC, so as to enable the RNG and Misc channels and disable the left and right channels, the max state index should be set as “2” by setting afe_1P2V_reg114<6:4> as 0x2.

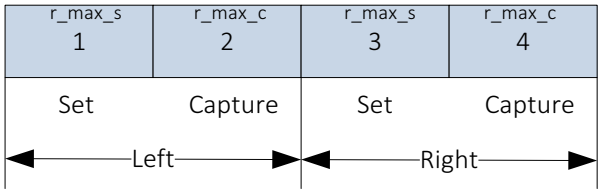
The total duration is (2*r_max_mc+1*r_max_s) * 24M clock period.



11.3.4.8 Case 8: 2-channel sampling for stereo audio

In this case, afe_1P2V_reg114<3:0> should be set as 0x3, so as to enable theleft and right channels and disable theRNG and Misc channels, the max state index should be set as “4” by setting afe_1P2V_reg114<6:4> as 0x4.

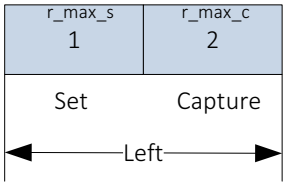
The total duration is (2*r_max_s+2*r_max_c) * 24M clock period.



11.3.4.9 Case 9: 1-channel sampling for mono audio

In this case, afe_1P2V_reg114<3:0> should be set as 0x1, so as to enable theleftchannel and disable theRNG, right andMisc channels, the max state index should be set as “2” by setting afe_1P2V_reg114<6:4> as 0x2.

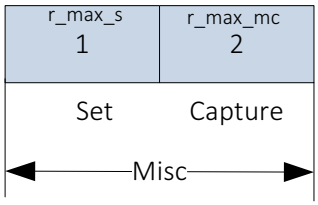
The total duration is (1*r_max_s+1*r_max_c) * 24M clock period.



11.3.4.10 Case 10: 1-channel sampling for Misc

In this case, afe_1P2V_reg114<3:0> should be set as 0x4, so as to enable the Miscchannel and disable theRNG, left and right channels, the max state index should be set as “2” by setting afe_1P2V_reg114<6:4> as 0x2.

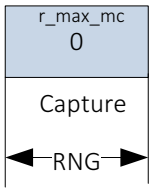
The total duration is (1*r_max_s+1*r_max_mc) * 24M clock period.



11.3.4.11 Case 11: 1-channel sampling for RNG

In this case, afe_1P2V_reg114<3:0> should be set as 0x8, so as to enable the RNG channel and disable the left, right andMisc channels, the max state index should be set as “0” by setting afe_1P2V_reg114<6:4> as 0x0.

The total duration is 1*r_max_mc * 24M clock period.



11.4 Register table

Table 11- 1 Register table related to SAR ADC

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg103<1:0>	adc_vrefl	00	Select V_{REF} for left channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: VBAT/N (N=2/3/4, when afe_1P2V_reg121<3:2> is set as 0x3/0x2/0x1).

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg103<3:2>	adc_vrefr	00	Select V_{REF} for right channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: VBAT/N (N=2/3/4, when afe_1P2V_reg121<3:2> is set as 0x3/0x2/0x1)
afe_1P2V_reg103<5:4>	adc_vrefm	00	Select V_{REF} for Misc channel 0x0: 0.6V 0x1: 0.9V 0x2: 1.2V 0x3: VBAT/N (N=2/3/4, when afe_1P2V_reg121<3:2> is set as 0x3/0x2/0x1)
afe_1P2V_reg103<7:6>	RSVD		
afe_1P2V_reg104<3:0>	adc_ain_m_n	0000	Select negative input for Misc channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output) 0xd: Ground 0xe: Ground 0xf: Ground
afe_1P2V_reg104<7:4>	adc_ain_m_p	0000	Select positive input for Misc channel: (For RNS mode, must set as 0x0.) 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output) 0xd: tempsensor_n (Temperature sensor negative output) 0xe: tempsensor_p (Temperature sensor positive output) 0xf: Vbat (Battery voltage)

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg105<3:0>	adc_ain_l_n	00	Select negative input for left channel 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output) 0xd: Ground 0xe: Ground 0xf: Ground
afe_1P2V_reg105<7:4>	adc_ain_l_p	00	Select positive input for left channel: (For RNS mode, must set as 0x0) 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output) 0xd: tempsensor_n (Temperature sensor negative output) 0xe: tempsensor_p (Temperature sensor positive output) 0xf: Vbat (Battery voltage)
afe_1P2V_reg106<3:0>	adc_ain_r_n	00	Select negative input for right channel: 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_n<0> (PGA left-channel negative output) 0xc: pga_n<1> (PGA right-channel negative output) 0xd: Ground 0xe: Ground 0xf: Ground

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg106<7:4>	adc_ain_r_p	0000	Select positive input for right channel: (For RNS mode, must set as 0x0) 0x0: No input 0x1: B<0> 0x2: B<1> ... 0x8: B<7> 0x9: C<4> 0xa: C<5> 0xb: pga_p<0> (PGA left-channel positive output) 0xc: pga_p<1> (PGA right-channel positive output) 0xd: tempsensor_n (Temperature sensor negative output) 0xe: tempsensor_p (Temperature sensor positive output) 0xf: Vbat (Battery voltage)
afe_1P2V_reg107<1:0>	adc_resl	11	Set resolution for left channel If RNS mode is set, resolution is set to 1. Otherwise ADC mode setting is: 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_1P2V_reg107<3:2>	RSVD		
afe_1P2V_reg107<5:4>	adc_resr	11	Set resolution for right channel If RNS mode is set, resolution is set to 1bit. Otherwise ADC mode setting is: 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_1P2V_reg107<7:6>	RSVD		
afe_1P2V_reg108<1:0>	adc_resm	11	Set resolution for Misc channel If RNS mode is set, resolution is set to 1. Otherwise ADC mode setting is: 0x0: 8bits 0x1: 10bits 0x2: 12bits 0x3: 14bits
afe_1P2V_reg108<3:2>	RSVD		
afe_1P2V_reg108<4>	adc_en_diffl	0	Select single-end or differential input mode for left channel. 0: single-ended mode 1: differential mode

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg108<5>	adc_en_diffr	0	Select single-end or differential input mode for right channel. 0: single-ended mode 1: differential mode
afe_1P2V_reg108<6>	adc_en_diffm	0	Select single-end or differential input mode for Misc channel. 0: single-ended mode 1: differential mode
afe_1P2V_reg108<7>	RSVD		
afe_1P2V_reg109<3:0>	adc_tsampl	0000	Number of ADC clock cycles in sampling phase for left channel to stabilize the input before sampling: 0x0: 3 cycles (RNS mode default) 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles … 0xf: 48 cycles
afe_1P2V_reg109<7:4>	adc_tsampr	0000	Number of ADC clock cycles in sampling phase for right channel to stabilize the input before sampling: 0x0: 3 cycles (RNS mode default) 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles … 0xf: 48 cycles
afe_1P2V_reg110<3:0>	adc_tsampm	0000	Number of ADC clock cycles in sampling phase for Misc channel to stabilize the input before sampling: 0x0: 3 cycles (RNS mode default) 0x1: 6 cycles 0x2: 9 cycles 0x3: 12 cycles … 0xf: 48 cycles
afe_1P2V_reg111<7:0>	r_max_mc[7:0]		r_max_mc[9:0]serves to set length of “capture” state for RNS and Misc channel. r_max_c[9:0] serves to set length of “capture” state for left and right channel. r_max_s serves to set length of “set” state for left, right and Misc channel. Note: State length indicates number of 24M clock cycles occupied by the state.
afe_1P2V_reg112<7:0>	r_max_c[7:0]		
afe_1P2V_reg113<3:0>	r_max_s		
afe_1P2V_reg113<5:4>	r_max_c[9:8]		
afe_1P2V_reg113<7:6>	r_max_mc[9:8]		
afe_1P2V_reg114<0>	r_en_left	0	Enable left channel. 1: enable
afe_1P2V_reg114<1>	r_en_right	0	Enable right channel. 1: enable

Address Decimal (Base address: 0x80)	Mnemonic	Default value	Description
afe_1P2V_reg114<2>	r_en_misc		Enable Misc channel sampling. 1: enable
afe_1P2V_reg114<3>	r_en_rns	0	Enable RNS sampling. 1: enable
afe_1P2V_reg114<6:4>	r_max_scnt	00	Set total length for sampling state machine (i.e. max state index)
afe_1P2V_reg114<7>	rsvd		
afe_1P2V_reg115<7:0>	rsvd		
afe_1P2V_reg116<2:0>	adc_clk_div	011	ADC clock (derive from external 24M crystal) ADC clock frequency = 24M/(adc_clk_div+1)
afe_1P2V_reg116<7:3>	rsvd		
afe_1P2V_reg117<7:0>	rng[7:0]		Read only, random number [7:0]
afe_1P2V_reg118<7:0>	rng[15:8]		Read only, random number [15:8]
afe_1P2V_reg119<7:0>	adc_dat[7:0]		Read only, Miscadcdat[7:0]
afe_1P2V_reg120<7:0>	adc_dat[15:8]		Read only [7]: vld, ADC data valid status bit (This bit will be set as 1 at the end of capture state to indicate the ADC data is valid, and will be cleared when set state starts.) [6:0]: Miscadc_dat[14:8]
afe_1P2V_reg121<3:2>	adc_vbat_div	00	Vbat divider select (see adc_vref) 0x0: OFF 0x1: VBAT/4 0x2: VBAT/3 0x3: VBAT/2
afe_1P2V_reg122<7:6>	adc_sel_ai_scale	0	Analog input pre-scaling select sel_ai_scale[1:0]: scaling factor 0x0: 1 0x1: 1/2 0x2: 1/4 0x3: 1/8
afe_1P2V_reg124<4>	adc_mode	0	0: normal mode 1: rsvd (RNS mode)
afe_1P2V_reg124<5>	adc_pd	1	Power down ADC 1: Power down 0: Power up

12 PGA

The ST17H58 integrates a PGA (Programmable Gain Amplifier) module.

The PGA supports two channels including left channel and right channel, and each channel consists of Boost stage pre-amplifier and Gain stage post-amplifier.

The PGA is used in combination with the audio and ADC module: By adjusting the gain of pre-amplifier and post-amplifier, the PGA can amplify mono or stereo differential input audio signals from specific AMIC pins before ADC sampling.

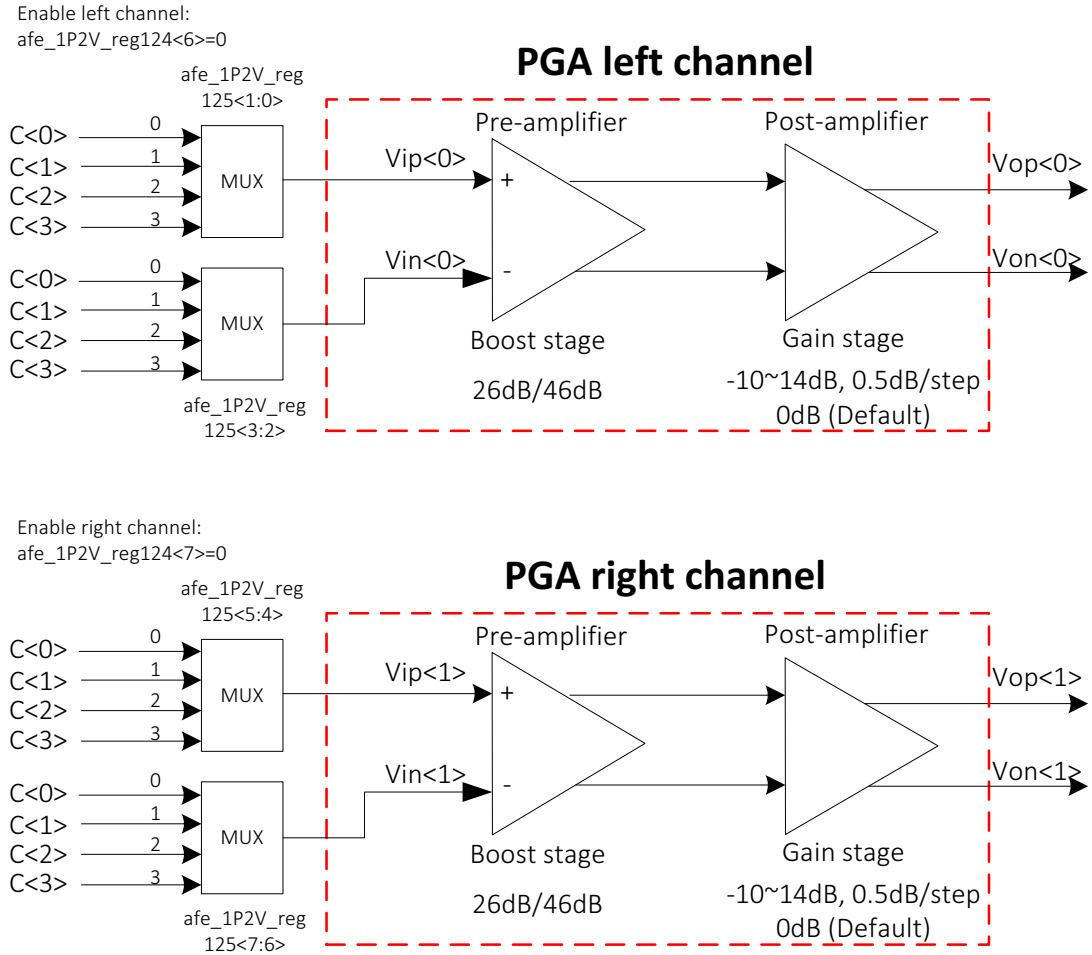


Figure 12- 1 Block diagram of PGA

***Note:**

Vip<0>, Vin<0>: Positive/Negative input of PGA left channel;

Vop<0>, Von<0>: Positive/Negative output of PGA left channel;

Vip<1>, Vin<1>: Positive/Negative input of PGA right channel;

Vop<1>, Von<1>: Positive/Negative output of PGA right channel.

12.1 Power on/down

Both PGA left channel and right channel are disabled by default.

To power on PGA left/right channel, the analog register pga_pd_l (afe_1P2V_reg124<6>) / pga_pd_r (afe_1P2V_reg124<7>) should be set as 1b'0.

12.2 Select input channel

When PGA left channel is enabled, the positive and negative input for the left channel are selectable from ANA_C<0>~ANA_C<3> via the analog register pga_sel_vin_l_p (afe_1P2V_reg125<1:0>) / pga_sel_vin_l_n (afe_1P2V_reg125<3:2>).

When PGA right channel is enabled, the positive and negative input for the right channel are selectable from ANA_C<0>~ANA_C<3> via the analog register pga_sel_vin_r_p (afe_1P2V_reg125<5:4>) / pga_sel_vin_r_n (afe_1P2V_reg125<7:6>).

12.3 Adjust gain

To adjust the PGA gain, the ALC of the audio module should be configured as analog mode, and the following three cases can apply:

1) **Auto regulate**

If auto regulate function is enabled, user can set an initial PGA gain. The (AMIC input * current PGA gain) is compared with the pre-configured high volume target, low volume target, volume noise level and VAD noise threshold, and the PGA gain will be automatically adjusted within the pre-configured range accordingly.

Please refer to section**9.2.5.1 Auto regulate in analog mode** for details.

2) **Manual mode 1**

If manual mode 1 is enabled, the PGA gain is directly adjustable via digital register 0xb63: Address 0xb63[6] serves to set gain for the pre-amplifier as 26dB or 46dB; while address 0xb63[5:0] serves to set gain for the post-amplifier as -10dB (0x0) ~ 14dB (0x30) with step of 0.5dB. The total PGA gain should be the sum of the two gain values.

Please refer to section **9.2.5.2 Manual regulate in analog mode** for details.

3) **Manual mode 2**

If manual mode 1 is enabled, the PGA gain will be automatically adjusted to the pre-configured target gain value with the pre-configured speed.

Please refer to section **9.2.5.2 Manual regulate in analog mode** for details.

12.4 Register table

Table 12- 1 Analog register table related to PGA

Address	Mnemonic	Default	Description
afe_1P2V_reg124<6>	pga_pd_l	1	Power down left channel PGA 1: Power down, 0: Power up
afe_1P2V_reg124<7>	pga_pd_r	1	Power down right channel PGA 1: Power down, 0: Power up
afe_1P2V_reg125<1:0>	pga_sel_vin_l_p	00	Select left channel positive input source. Gate off all input with pga_pd_l. 0x0: C<0> 0x1: C<1> 0x2: C<2> 0x3: C<3>
afe_1P2V_reg125<3:2>	pga_sel_vin_l_n	00	Select left channel negative input source. Gate off all input with pga_pd_l. 0x0~0x3: C<0>~ C<3>
afe_1P2V_reg125<5:4>	pga_sel_vin_r_p	00	Select right channel positive input source. Gate off all input with pga_pd_r. 0x0~0x3: C<0>~ C<3>
afe_1P2V_reg125<7:6>	pga_sel_vin_r_n	00	Select right channel negative input source. Gate off all input with pga_pd_r. 0x0~0x3: C<0>~ C<3>

For digital registers related to the PGA, please refer to section **9.2.5**and **9.2.6**.

13 Temperature Sensor

The ST57H25F512integrates a temperature sensor and it's used in combination with the SAR ADC to detect real-time temperature.

The temperature sensor is disabled by default. The analog register afe3V_reg07<4>should be set as 1b'0 to enable the temperature sensor.

Table 13- 1 Analog register for temperature sensor

Address	Name	Description	Default Value
afe3V_reg07<4>	pd_temp_sensor_3V	Power on/down temperature sensor: 0: Power up 1: Power down	1

The temperature sensor embeds two diodes. It takes the real-time temperature (T) as input, and outputs two-way forward voltage drop (V_{BE}) signals of diodes as positive and negative output respectively.

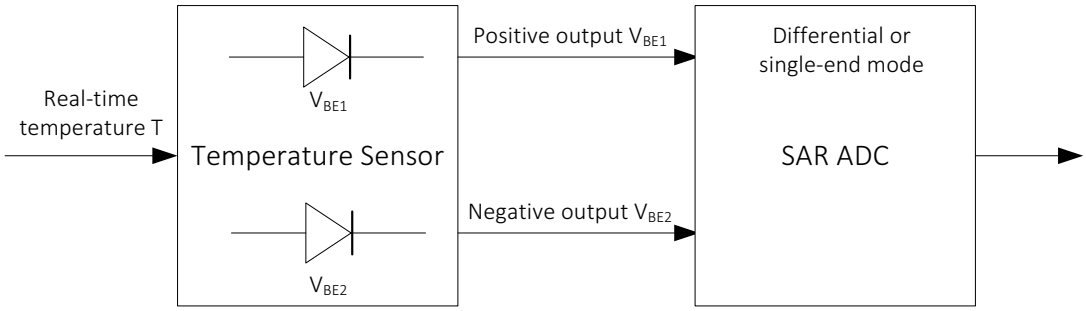


Figure 13- 1 Block diagram of temperature sensor

The difference of the two-way V_{BE} signals(ΔV_{BE}) is determined by the real-time temperature T, as shown below:

$$\Delta V_{BE} = 130mV + 0.51mV/℃ * \left(T - (-40℃)\right)$$
$$= 130mV + 0.51mV/℃ * (T + 40℃)$$

In this formula, "130mV" indicates the value of ΔV_{BE}at the temperature of "$-40℃$".

To detect the temperature, the positive and negative output of the temperature sensor should be enabled as the input channels of the SAR ADC. The ADC will convert the two-way V_{BE} signals into digital signal.

- ✧ If the ADC is configured as single-end mode, the ADC should initiate two operations, each time the positive/negative output of the temperature sensor should be configured as the single input of the ADC, and two output signals (V_1 and V_2) are obtained, therefore,

$$\Delta V_{BE} = \frac{(V_1 - V_2)}{2^N - 1} * V_{REF}.$$

In the two formulas, “N” and “V_{REF}” indicates the selected resolution and reference voltage of the SAR ADC.

✧ If the ADC is configured as differential mode,the positive and negative output of the temperature sensor should be configured as differential input of the ADC. The ADC should initiate one operation and obtain one output signal (ADCOUT), therefore,

$$\Delta V_{BE} = \frac{ADCOUT}{2^N - 1} * V_{REF}.$$

Then the real-time temperature T can be calculated according to the ΔV_{BE}.

14 Low Power Comparator

The ST17H58 embeds a low power comparator. This comparator takes two inputs: input derived from external PortA (ANA_A<0>~ANA_A<7>), and reference input derived from internal reference, ANA_A<0>, ANA_A<3>, AVDD3 or float.

By comparing the input voltage multiplied by selected scaling coefficient with reference input voltage, the low power comparator will output high or low level accordingly.

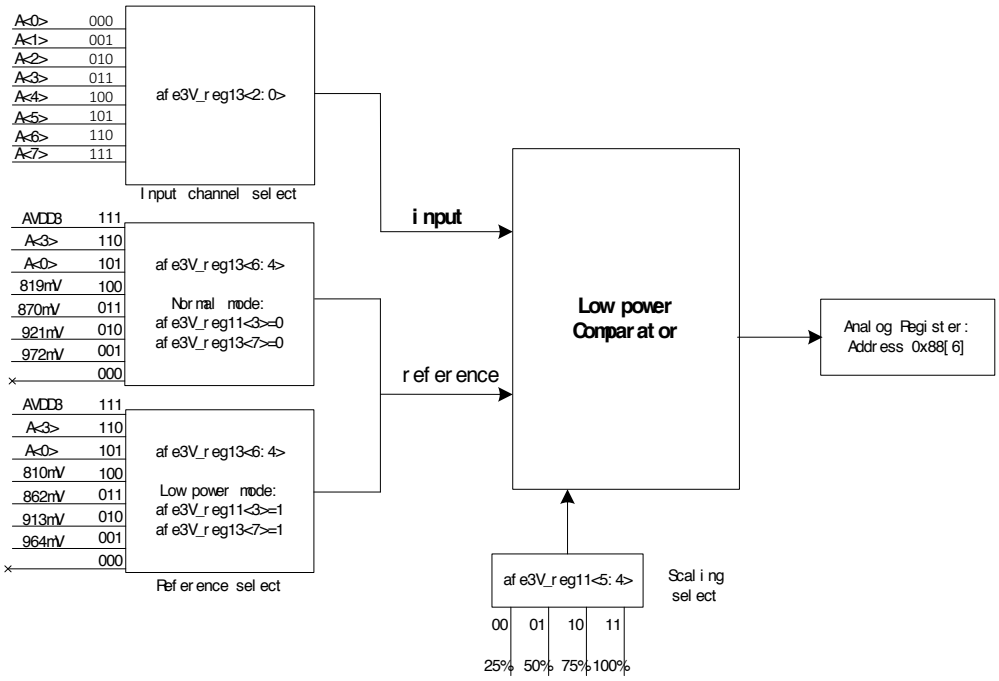


Figure 14- 1 Block diagram of low power comparator

14.1 Power on/down

The low power comparator is powered down by default.

The analog register afe3V_reg07<3>serves to control power state of the low power comparator: By clearing this bit, this comparator will be powered on; by setting this bit to 1b'1, this comparator will be powered down.

To use the low power comparator, first set afe3V_reg07<3> as 1b'0, then the 32K RC clock source is enabled as the comparator clock.

14.2 Select input channel

Input channel is selectable from the PortA (ANA_A<0>~ANA_A<7>) via the analog register afe3V_reg13<2:0>.

14.3 Select mode and input channel for reference

Generally, it's needed to clear both the afe3V_reg11<3>and afe3V_reg13<7> to select the

normal mode. In normal mode, the internal reference is derived from UVLOandhas higher accuracy, but current bias is larger (10uA); reference voltage input channel is selectable from internal reference of 972mV, 921mV, 870mV and 819mV, as well as ANA_A<0>, ANA_A<3>, AVDD3 and float.

To select the low power mode, both the afe3V_reg11<3> and afe3V_reg13<7> should be set as 1b'1. In low power mode, the internal reference is derived from Bandgapand haslower accuracy, but current bias is decreased to 50nA; reference voltage input channel is selectable from internal reference of 964mV, 913mV, 862mV and 810mV, as well as ANA_A<0>, ANA_A<3>, AVDD3 and float.

14.4 Select scaling coefficient

Equivalent reference voltage equals the selected referenceinputvoltage divided by scaling coefficient.

The analog register afe3V_reg11<5:4> serves to select one of the four scaling options: 25%, 50%, 75% and 100%.

14.5 Low power comparator output

The low power comparatoroutput is determined by the comparison result of the value of [input voltage *scaling] andreference voltage input. The comparison principleis shown as below:

- ✧ If the value of [input voltage *scaling] is larger than reference voltage input, the output will be low ("0").
- ✧ If the value of [input voltage *scaling] is lower than reference voltage input, the output will be high ("1").
- ✧ If the value of [input voltage *scaling] equalsreference voltage input, or input channel is selected as float, the output will be uncertain.

User can readthe output of the low power comparator viatheanalog registerafe_1P2V_reg08<6> (0x88[6]).

The output of the low power comparator can be used as signalto wakeup system from low power modes.

14.6 Register table

Table 14- 1 Analogregister table related to low power comparator

Address	Name	Description	Default Value
afe3V_reg07<3>	pd_lc_comp_3V	Power on/down low power comparator: 0: Power up 1: Power down	1
afe3V_reg11<3>	ref_mode_sel	Reference mode select: 0: normal mode (current bias 10uA) 1: low power mode (current bias 50nA)	1

Address	Name	Description	Default Value
		See afe3V_reg13<7>.	
afe3V_reg11<5:4>	comp_refscale<1:0>	Reference voltage scaling: 00: 25% 01: 50% 10: 75% 11: 100%	01
afe3V_reg13<2:0>	comp_chsel<2:0>	Input Channel select: 000: A<0> 001: A<1> 010: A<2> 011: A<3> 100: A<4> 101: A<5> 110: A<6> 111: A<7>	000
afe3V_reg13<3>	vbus_inen	rsvd	0
afe3V_reg13<6:4>	comp_refesel<2:0>	Reference select: normal mode low power mode 000: Float 000: Float 001: 972mV 001: 964mV 010: 921mV 010: 913mV 011: 870mV 011: 862mV 100: 819mV 100: 810mV 101: A<0>101: A<0> 110: A<3>110: A<3> 111: AVDD3111: AVDD3	000
afe3V_reg13<7>	pd_I10u	Enable or disable 10uA current bias: 0: Enable 10uA current bias 1: Disable 10uA current bias	1

15 Continuous Comparator

15.1 Function overview

The ST17H58 embeds a continuous comparator, which is powered down by default. The analog register afe1V_reg06<4> serves to enable this comparator core.

This continuous comparator takes two inputs: VIN+input(0V~VDD) derived from external PortA (ANA_A<0>~ANA_A<7>), and VIN- input derived from multiple reference sources.

- The analog register afe1v_reg03<5:3> serves to select one analog pin from PortA as VIN+input.
- VIN- input depends on selected operation mode of the continuous comparator.
- The continuous comparator supports two operation modes, including single-end mode and differential mode. Each mode supports three sub-modes with different response speed and power consumption, i.e. normal mode, low power mode and high speed mode. High speed mode takes less response time but more power consumption, while low power mode takes less power consumption but more response time.
- The analog register afe1v_reg03<2:0> serves to select operation mode, as well as speed and power consumption mode.

By comparing the VIN+inputwithVIN- input, the continuous comparator will output high or low level accordingly.

- If the VIN+input is lower than VIN- input, the output will be low (“0”).
- If the VIN+input is larger than VIN- input, the output will be high (“1”).
- If the VIN+inputequalsVIN-input, the output will be uncertain.

The output of the continuous comparator can be used for interrupt handling.

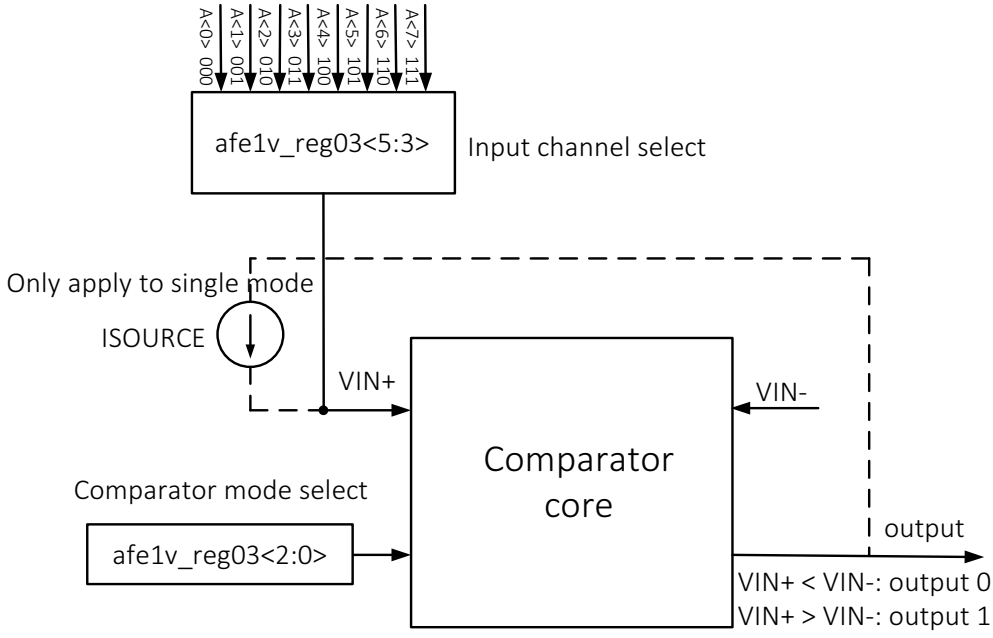


Figure 15- 1 Block diagram of continuous comparator

In single-end mode, an ISOURCE module can be enabled to create a feedback path between the comparator output and the selected VIN+ input. It forms a relaxation oscillator, which will sink current from or source current into VIN+ according to comparator output.

15.2 Differential mode

If afe1v_reg03<2:0> is set as 3b'1xx (100/101/110), differential mode is selected. In this mode, the analog register afe1V_reg05<6> serves to select either ANA_A<0> or ANA_A<1> as VIN-input. Before the comparator core is enabled, afe1V_reg03<5:0> and afe1V_reg05<6> must be configured.

In this mode, a hysteresis (disabled by default) can be enabled via the analog register afe1V_reg05<7>. To achieve 40mV or so differential hysteresis ($V_{DIFFHYST}$≈40mV), the analog register afe1V_reg04<7:6>should be configured according to the value of VDD. The VIN+ input is compared with (base voltage ±$V_{DIFFHYST}$/2) and will output accordingly:

- ✧ Base voltage is relevant to VIN- and VDD.
- ✧ If the VIN+input is lower than (base voltage - $V_{DIFFHYST}$/2), the output will be low ("0").
- ✧ If the VIN+input is larger than (base voltage + $V_{DIFFHYST}$/2), the output will be high ("1").
- ✧ If the VIN+inputis among the range of the (base voltage - $V_{DIFFHYST}$/2) ~ (base voltage + $V_{DIFFHYST}$/2), the output will retain the previous output.

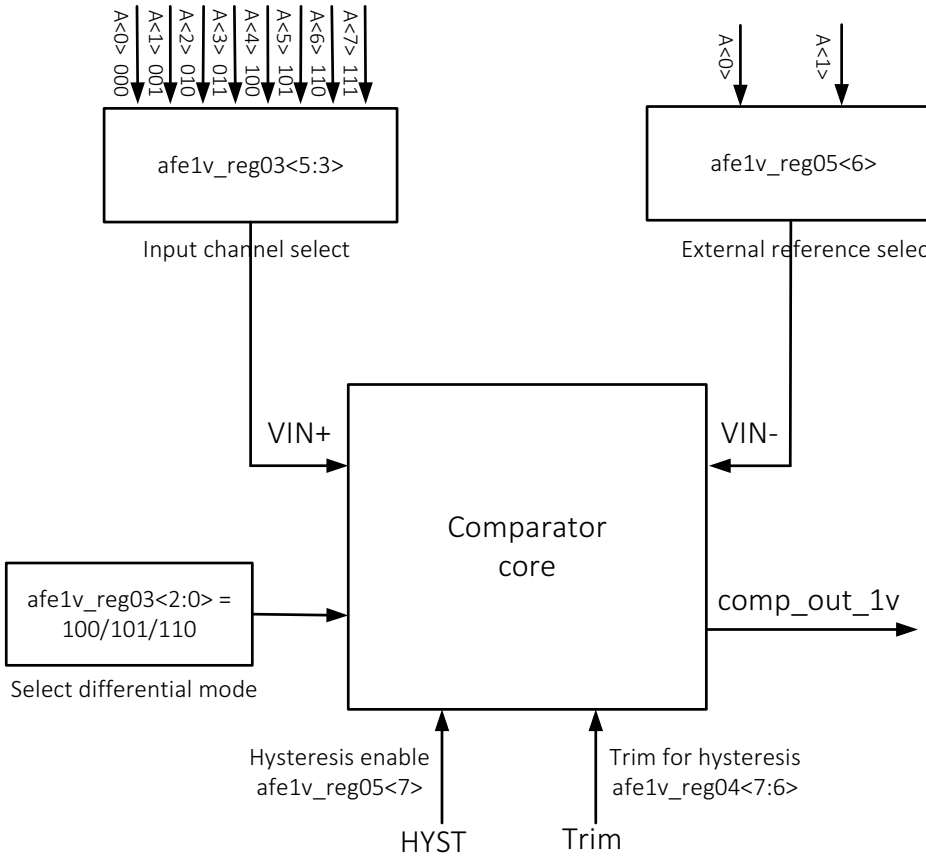


Figure 15- 2 Continuous comparator in differential mode

15.3 Single-end mode

If afe1v_reg03<2:0> is set as 3b'0xx (000/001/010), single-end mode is selected. In this mode,VIN-input is derived from a reference ladder. The reference ladder generates two voltage references "VUP" and "VDOWN" based on internal reference voltage "VREF", and VIN- will switch to VUP or VDOWN according to the comparison result of VIN+ and VIN-.

- The reference ladder is disabled by default.Whenthe reference ladderis powered down, VUP, VDOWN and VIN- are all 0V. To enable the ladder, afe1V_reg06<5> should be set as 1b'0.
- VREF is selectable from VDD, external reference "AREF" or internal 1.2V/1.8V/2.4V reference via the afe1v_reg06<2:0>.
- "AREF" is selectable as either ANA_A<0> or ANA_A<1> via the afe1v_reg05<6>.
- "VUP" and "VDOWN" are configurable via the afe1V_reg04<5:0> and afe1V_reg05<5:0>.
- When it's detected that VIN+ is larger than VIN-, VIN- will switch to VDOWN. When VIN+ drops below VIN- again, VIN- will be switched back to VUP.

Before the comparator core is enabled, afe1V_reg03<5:0>, afe1V_reg04<5:0>, afe1V_reg05<6:0>, afe1V_reg06<2:0> must be configured.

In single-end mode, the ISOURCE (disabled by default) can be enabled via afe1V_reg06<3> to form a relaxation oscillator.

- In this case, the selected VIN+ needs to be connected to GND via an external capacitive sensor.
- The current value of the ISOURCE is selectable from 0, 2.5uA, 5uA or 10uA via afe1V_reg03<7:6>.
- The frequency of this oscillator,f_osc,equals I_SOURCE/(2C* (VUP-VDOWN)). "C" indicates capacitance value of the external capacitor connected with the selected VIN+; "I_SOURCE" indicates the selected current value of the ISOURCE.
- The oscillator circuit will sink current from VIN+ when the comparator output is high, and source current into VIN+ when the comparator output is low.

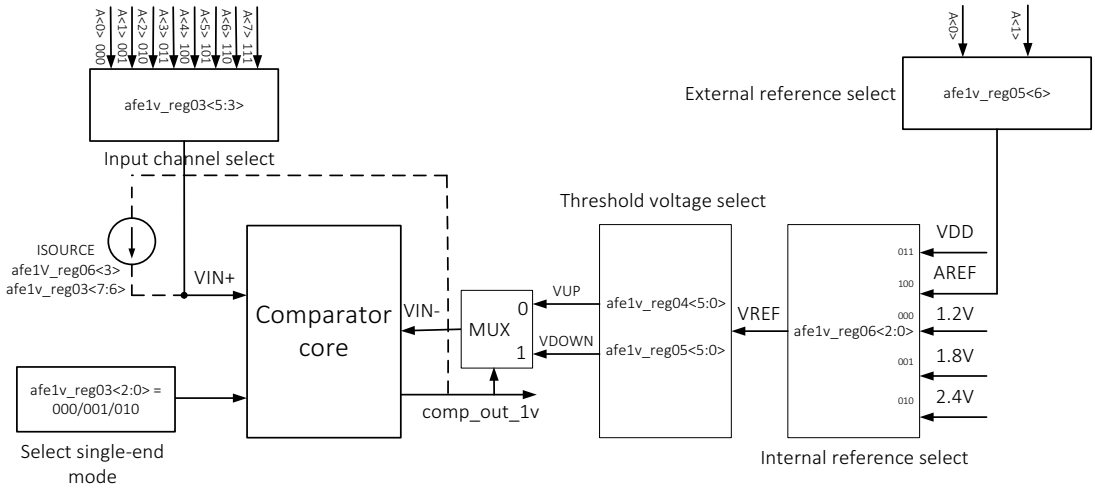


Figure 15- 3 Continuous comparator in single-end

15.4 Register table

Table 15- 1 Analog register table related to continuous comparator

Address	Name	Description	Default Value
afe1V_reg03<2:0>	mode<2:0>	Comparator mode: 000: single low power 001: single normal 010: single high speed 100: differential low power 101: differential normal 110: differential high speed	000
afe1V_reg03<5:3>	psel<2:0>	Select analog pin as VIN+ input: 000: A<0> 001: A<1> 010: A<2> 011: A<3> 100: A<4> 101: A<5> 110: A<6> 111: A<7>	000
afe1V_reg03<7:6>	isc<1:0>	Isource current value: 00 : 0.0 uA 01 : 2.5 uA 10 : 5.0 uA 11 : 10.0 uA	00
afe1V_reg04<5:0>	thup<5:0>	Threshold configuration for hysteresis unit: VUP = (thup+1)/64 * VREF	100001
afe1V_reg04<7:6>	trim<1:0>	Trim for differential mode comparator hysteresis(40mV): VDD 3.0 ~ 3.6 V: 00 VDD 2.5 ~ 3.0 V: 01 VDD 2.1 ~ 2.5 V: 10 VDD 1.8 ~ 2.1 V: 11	00
afe1V_reg05<5:0>	thdn<5:0>	Threshold configuration for hysteresis unit: VDOWN=(thdn+1)/64* VREF	011111
afe1V_reg05<6>	extrefsel	External analog reference select (AREF): 0: use A<0> as external analog reference 1: use A<1> as external analog reference	0
afe1V_reg05<7>	hyst	Comparator hysteresis 0: disable 1: enable	0
afe1V_reg06<2:0>	refsel<2:0>	Internal analog reference select: 000: VREF = internal 1.2 V reference 001: VREF = internal 1.8 V reference 010: VREF = internal 2.4 V reference 011: VREF = VDD 100: VREF = AREF	011
afe1V_reg06<3>	isource_pd	Isource power down:	1

Address	Name	Description	Default Value
		0: enable 1: disable	
afe1V_reg06<4>	comp_pd	Comparator core power down: 0: enable 1:disable	1
afe1V_reg06<5>	ladder_pd	Reference ladder resistance power down: 0: enable 1: disable	1

16 AES

The **ST57H25**F512embeds AES module with encryption and decryption function. The input 128bit plaintext in combination of key is converted into the final output ciphertext via encryption; the 128bit ciphertext in combination of key can also be converted into 128bit plaintext via decryption.

The AES hardware accelerator provides automatic encryption and decryption. It only takes (1000*system clock cycles) to implement AES encryption/decryption. Suppose system clock is 20MHz, the time needed for AES encryption/decryption is 50us.

Both RISC mode and DMA mode are supported for AES operation.

16.1 RISC mode

For RISC mode, configuration of related registers is as follows:

1) Set the value of key via writing registers AES_KEY0~ AES_KEY15(address 0x550~0x55f).
2) Set operation method of AES module via register AES_CTRL: set address 0x540[0] as 1b'1 for decryption method, while clear this bit for encryption method.
3) For encryption method, write registers AES-DAT0~ AES-DAT3 (address 0x548~0x54b) for four times to set the 128bit plaintext. After encryption, the 128bit ciphertext can be obtained by reading address 0x548~0x54b for four times.
4) For decryption method, write registers AES-DAT0~ AES-DAT3 (address 0x548~0x54b) for four times to set the 128bit ciphertext. After decryption, the 128bit plaintext can be obtainedby reading address 0x548~0x54b for four times.
5) Address 0x540 bit[1] and bit[2] are read only bits: bit[1] will be cleared automatically after quartic writing of address 0x548~0x54b; bit[2] will be set as 1 automatically after encryption/decryption, and then cleared automatically after quartic reading of address 0x548~0x54b.

16.2 DMA mode

As for DMA mode, it is only needed to configure the value of key and encryption/decryption method for AES module. Please refer to point 1) ~ 2) in section**16.1**.

16.3 AES-CCM

The AES-CCM(Counter with the CBC-MAC) mode is disabled by default. AES output is directly determined by current encryption and decryption, irrespective of previous encryption and decryption result.

If 0x540[7] is set as 1b'1 to enable AES-CCM mode, AES output will also take previous encryption and decryption result into consideration.

16.4 Register table

Table 16- 1 Register table related to AES

Address	Mnemonic	Type	Description	Reset Value
0x540	AES_CTRL	R/W	[0]Select decrypt/encrypt. 1: decrypt, 0: encrypt [1] Read-only. 1: input data needed, 0: input data ready. [2]Read-only. 0: output data not ready, 1:output data ready. [7]1: enable AES-CCM mode.	00
0x548	AES-DAT0		Input/Output Data byte 0	
0x549	AES-DAT1		Input/Output Data byte 1	
0x54a	AES-DAT2		Input/Output Data byte 2	
0x54b	AES-DAT3		Input/Output Data byte 3	
0x550	AES_KEY0	R/W	[7:0] KEY0	00
0x551	AES_KEY1	R/W	[7:0] KEY1	00
0x552	AES_KEY2	R/W	[7:0] KEY2	00
0x553	AES_KEY3	R/W	[7:0] KEY3	00
0x554	AES_KEY4	R/W	[7:0] KEY4	00
0x555	AES_KEY5	R/W	[7:0] KEY5	00
0x556	AES_KEY6	R/W	[7:0] KEY6	00
0x557	AES_KEY7	R/W	[7:0] KEY7	00
0x558	AES_KEY8	R/W	[7:0] KEY8	00
0x559	AES_KEY9	R/W	[7:0] KEY9	00
0x55a	AES_KEY10	R/W	[7:0] KEY10	00
0x55b	AES_KEY11	R/W	[7:0] KEY11	00
0x55c	AES_KEY12	R/W	[7:0] KEY12	00
0x55d	AES_KEY13	R/W	[7:0] KEY13	00
0x55e	AES_KEY14	R/W	[7:0] KEY14	00
0x55f	AES_KEY15	R/W	[7:0] KEY15	00

17 Key Electrical Specifications

Note: The electrical characteristics currently listed in this section are target specifications and only supplied for reference. Some data may be updated according to actual test results.

17.1 Absolute maximum ratings

Table 17- 1 Absolute Maximum Ratings

Characteristics	Sym.	Min.	Max	Unit	Test Condition
Supply Voltage	VDD	-0.3	3.9	V	All AVDD and DVDD pin must have the same voltage
Voltage on Input Pin	V_{In}	-0.3	VDD+ 0.3	V	
Output Voltage	V_{Out}	0	VDD	V	
Storage temperature Range	T_{Str}	-65	150	°C	
Soldering Temperature	T_{Sld}		260	°C	

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

17.2 Recommended operating condition

Table 17- 2 Recommended operation condition

Item	Sym.	Min	Typ.	Max	Unit	Condition
Power-supply voltage	VDD	1.8	3.3	4.3	V	
Operating Temperature Range	T_{Opr}	-40		85	°C	ET versions
		-40		125	°C	AT versions

17.3 DC characteristics

Table 17- 3 DC characteristics (T=25℃)

Mode	Sym.	Min	Typ.	Max	Unit	Condition
Tx	I_{Tx}		4.05		mA	Continuous TX transmission, 0dBm output power
			6.2		mA	5dBm output power
Rx	I_{Rx}		3.8		mA	Continuous Rx reception
			4.25		mA	Boost Mode
Suspend	I_{Susp}		TBD		uA	IO wakeup
	I_{Susp}		TBD		uA	Timer wakeup
Deep sleep with SRAM retention	I_{Deep1}		1.4		uA	
Deep sleep without SRAM retention			0.4		uA	
Shutdown			60		nA	

17.4 AC characteristics

Table 17- 4 AC Characteristics

Item	Sym.	Min	Typ.	Max	Unit	Condition
Digital inputs/outputs						
Input high voltage	VIH	0.7VDD		VDD	V	
Input low voltage	VIL	VSS		0.3VDD	V	
Output high voltage	VOH	VDD-0.3		VDD	V	
Output low voltage	VOL	VSS		0.3	V	
USB characteristics						
USB Output Signal Cross-over Voltage	V_{Crs}	1.3	-	2.0	V	
RF performance						
Item		**Min**	**Typ**	**Max**	**Unit**	
RF frequency range		2380		2500	MHz	Programmable in 1MHz step
Data rate	1Mbps, ±250KHz deviation 2Mbps, ±500KHz deviation 250Kbps, ±500KHz deviation 125Kbps, ±250KHz deviation 500Kbps, ±250KHz deviation					
BLE 1Mbps RF_Rxperformance(±250KHz deviation, 30%PER)						
Sensitivity	1Mbps	-94	-93	-91	dBm	
Frequency Offset Tolerance		-300		+300	KHz	
Co-channel rejection			-7		dB	Wanted signal at -67dBm
In-band blocking rejection	±1 MHz offset		12		dB	Wanted signal at -67dBm
	±2 MHz offset		33		dB	
	±3 MHz offset		35		dB	
	>4MHz offset		52		dB	
Image rejection			33		dB	Wanted signal at -67dBm
BLE 1Mbps RF_Tx performance						
Output power, maximum setting				8	dBm	supply power by lithium battery directly
				5	dBm	supply power by DCDC

Item	Sym.	Min	Typ.	Max	Unit	Condition
				10	dBm	Boost Mode
Output power, minimum setting				-22	dBm	supply power by lithium battery directly
				-25	dBm	supply power by DCDC
				-20	dBm	Boost Mode
Programmable output power range		30			dB	
Modulation 20dB bandwidth			1.3		MHz	
IEEE802.15.4 (250Kbps) RF_Rx performance(±500KHz deviation, 1%PER)						
Sensitivity	250Kbps		-97		dBm	
Frequency Offset Tolerance		-400		+400	KHz	
Co-channel rejection			-4		dB	Wanted signal at -67dBm
In-band blocking rejection	-2 MHz offset		6		dB	Wanted signal at -67dBm
	+2 MHz offset		6		dB	
	-3 MHz offset		19		dB	
	+3 MHz offset		19		dB	
	>4 MHz offset		28		dB	
Image rejection			28		dB	Wanted signal at -67dBm
IEEE802.15.4 (250Kbps) RF_Tx performance						
Output power, maximum setting			7	8	dBm	supply power by lithium battery directly
				5	dBm	supply power by DCDC
				10	dBm	Boost Mode
Output power, minimum setting			-23	-22	dBm	supply power by lithium battery directly
				-25	dBm	supply power by DCDC
				-20	dBm	Boost Mode
Programmable output power range		30			dB	
Modulation 20dB bandwidth			2.3		MHz	

Item	Sym.	Min	Typ.	Max	Unit	Condition
BLE 2Mbps RF_Rxperformance(±500KHz deviation, 1%PER)						
Sensitivity	2Mbps	-89	-88	-85	dBm	
Frequency Offset Tolerance		-200		+200	KHz	
Co-channel rejection			-7		dB	Wanted signal at -67dBm
In-band blocking rejection	±1 MHz offset		-7		dB	Wanted signal at -67dBm
	±2 MHz offset		10		dB	
	±3 MHz offset		23		dB	
	±4MHz offset		32		dB	
	>5MHz offset		52		dB	
Image rejection			30		dB	Wanted signal at -67dBm
BLE 2Mbps RF_Tx performance						
Output power, maximum setting			7	8	dBm	supply power by lithium battery directly
				5	dBm	supply power by DCDC
				10	dBm	Boost Mode
Output power, minimum setting			-23	-22	dBm	supply power by lithium battery directly
				-25	dBm	supply power by DCDC
				-20	dBm	Boost Mode
Programmable output power range		30			dB	
Modulation 20dB bandwidth			2.6		MHz	
Frequency synthesizer						
Phase noise, unmodulated carrier	±1MHz	-114		-108	dBc/ Hz	
	±2MHz	-122		-115	dBc/ Hz	
	±3MHz	-126		-119	dBc/ Hz	
RSSI						
RSSI range			45		dB	@BLE

Item	Sym.	Min	Typ.	Max	Unit	Condition
			55		dB	@Zigbee
Absoluteuncalibrated accuracy			1		dB	
24MHz crystal						
Nominal frequency (parallel resonant)	f_{NOM}		24		MHz	
Frequency tolerance	f_{TOL}	-20		+20	ppm	
Load capacitance	C_L	5	12	18	pF	Programmable on chip load cap
Equivalent series resistance	ESR		50	100	ohm	
32.768KHz crystal						
Nominal frequency (parallel resonant)	f_{NOM}		32.768		KHz	
Frequency tolerance	f_{TOL}	-100		+100	ppm	
Load capacitance	C_L	6	9	12.5	pF	Programmable on chip load cap
Equivalent series resistance	ESR		50	80	kohm	
24MHz RC oscillator						
Nominal frequency	f_{NOM}		24		MHz	
Frequency tolerance	f_{TOL}		1		%	On chip calibration
32kHz RC oscillator						
Nominal frequency	f_{NOM}		32		kHz	
Frequency tolerance	f_{TOL}		0.03		%	On chip calibration
Calibration time			3		ms	
ADC						
Differential nonlinearity	DNL		TBD		LSB	
Integral nonlinearity	INL		TBD		LSB	
Signal-to-noise and distortion ratio	SINAD		TBD		dB	fin=1kHz, fS=16kHz
Spurious free dynamic range	SFDR		TBD		dB	fin=1kHz, fS=16kHz
Effective Number of Bits	ENOB		TBD		bits	
Sampling frequency	Fs		TBD		KHz	