HIGHLY INTEGRATED EEG/ECG/EMG/PPG BIO ELECTRIC SENSOR RISC-V SOC

NNC-EPC001

Features

- Operating Voltage Range: 2.5V to 5V
- Operating Temperature Range: -40℃ to 125℃
- Support SIP with BLE DIE
- Power management
  - Low power modes: Sleep and Stop modes
- RISC-V core
  - 2-pipeline stage
  - support branch predictor and prefetch
  - support RV32I/E/M/A/C instruction
  - Dhrystone 1.41, Coremark 3.34
  - support 32-bit ILM and DLM
  - support icache
- Memories
  - 256kB on-chip FLASH
  - 64kB ROM
  - 64kB SRAM
- Digital Interfaces
  - Quad-SPI FLASH interface
  - 2 x UART, with hardware flow control
  - 2 x I2C, with hardware flow control
  - 2 x I2S, with hardware flow control
  - 3 x SPI interfaces
- Digital Function
  - 2 x General purpose timers
  - Dual timer & watchdog function
  - PWM, with hardware flow control
  - RTC, with hardware flow control
  - DMA, with hardware flow control
- EEG/ECG/EMG signal chain
  - Standalone ECG acquisition up to 32 kHz
  - Programmable INA gain: 1 to 100
  - Input noise: <2.5μVrms (BW=0.1~100HZ, GAIN=6)
  - PGA+24BITS ADC AFE
  - AC, DC lead-off detect
  - RLD driver
  - Digital filter:
    - -- HPF: 0.25/0.5/1/2.5/10/15/20/25Hz
    - -- LPF: 9-11/15-20/25/50/100/150/200/350Hz
    - -- Notch: 50Hz
- PPG receiver
  - 24-Bit Representation of the Current Input
  - Digital ambient subtraction at ADC output
  - Noise filtering with programmable bandwidth
  - Dynamic range up to 100 dB
- PPG transmitter
  - Four LEDs in common anode configuration
  - 8-Bit LED current up to 50 mA
  - Programmable LED on-time, duty and width
  - Support of two green LED for optimized HRM
  - Support the Red LED and IR LED for SpO2
  - Support of IR LED for Proximity sensor
- Heart APP inside the SoC
  - QT, LVET, PAT
  - SDNN, MSSD, SDANN, PNN50
  - Heart rate, respiration rate

Applications

- Brain signal detection system
- Synchronized PPG/ECG for blood pressure estimation
- HRM for wearables, hearables
- Heart-rate variability (HRV)
- Pulse oximetry (SpO2) measurements

Contents

List of Tables

List of Figures

1. Introduction

This datasheet provides characteristics and ordering information of the NNC-EPC001.

For information on the Starfive RISC-V core, please refer to the Starfive S2 Manual which available from the www.sifive.com website.

2. Description

The NNC_EPC001 is a specialized Bio-electric-sensing SOC. It is integrated with the highly accurate ECG/EEG/EMG/PPG Analog Front-End, which is suitable for bio-electric-sensor application. It integrates outstanding EEG/ECG/EMG/PPG features extraction capability using Heart APP, which makes it suitable for a wide range of wearable products working at low power consumption and high performance. The heart of this SoC is the processor RISC-V CORE(S2).

This powerful MCU and its internal memories make it possible to integrate with different wireless transceivers such as Bluetooth BLE or NFC for transferring measurement data. The NNC_EPC001 also has a rich peripheral interface which allows the SoC to be used with several different sensors.

2.1 Block Diagram

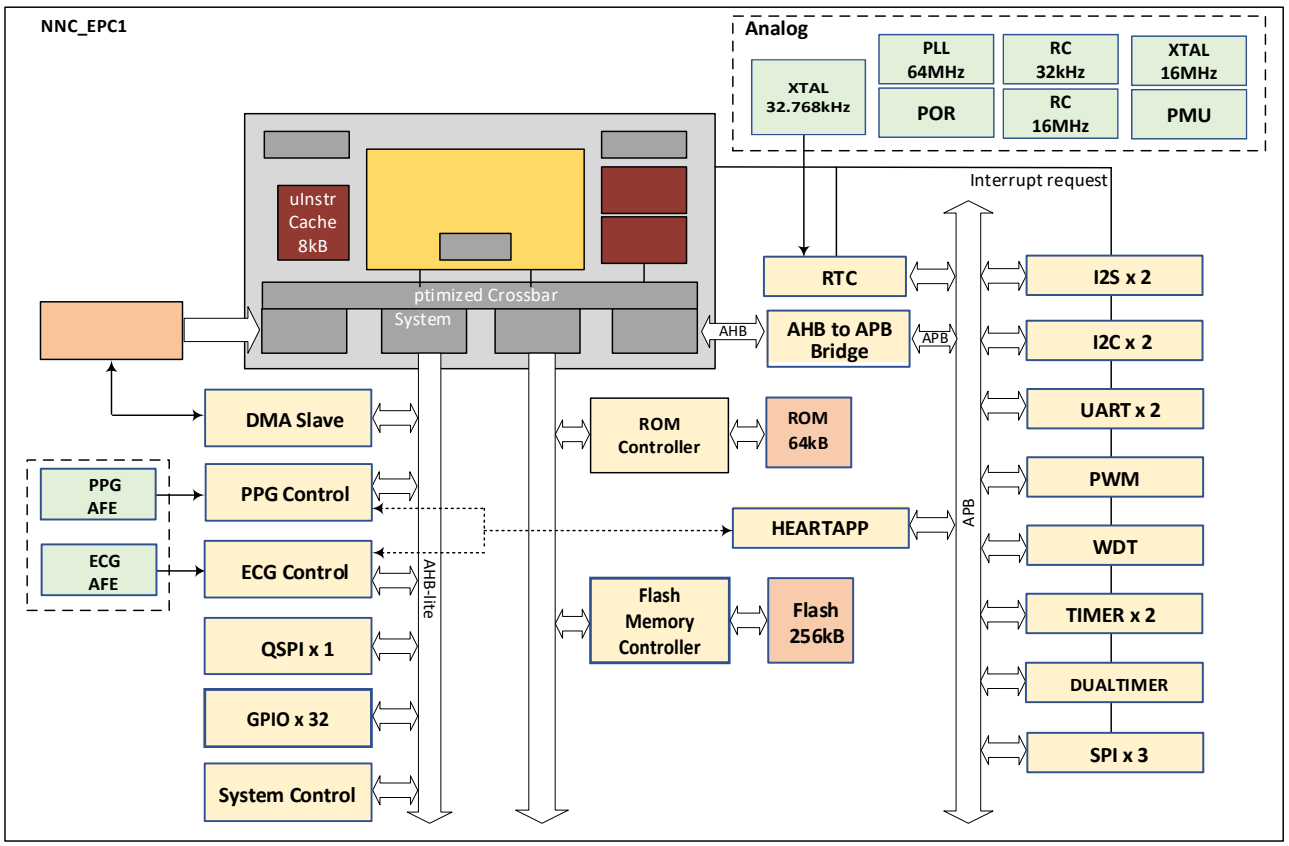


Figure 1 NNC-EPC001 Block diagram

3. Electrical Characteristics

3.1 Absolute maximum ratings

	MIN	MAX	UNIT
PAD_VBATA to PAD_AVSS	-0.3	5.5	V
VDDIO to VSSIO	-0.3	5.5	V
VDD_DIG to DVSS	-0.3	1.6	V
VDD_FLASH to VSS_FLASH	-0.3	1.6	V
Storage temperature, Tstg	-40	125	℃

Table 1 NNC_EPC001 Absolute maximum ratings

3.2 Recommended operating conditions

	MIN	NOM	MAX	UNIT
POWER SUPPLY				
Analog power supply (PAD_VBAT – PAD_AVSS)	2.7	3	5.5	V
Digital power supply (VDD_DIG)		1.5	1.6	V
IO supply (VDDIO - VSSIO)	2.7	3	5.5	V
FLASH supply (VDD_FLASH)		1.5	1.6	V
ANALOG INPUTS				℃
Full-scale differential input voltage range	-vref/gain		+vref/gain	V
Input current range	0.5		50	uA
CLOCK INPUT				
External clock input frequency		32		KHz
TEMPERATURE RANGE				
Operating temperature range	-40		85	℃

Table 2 NNC_EPC001 Recommended operating conditions

3.3 ESD Ratings

		VALUE	UNIT	
Electrostatic discharge performance		+/- 2000	V	

Table 3 NNC_EPC001 ESD ratings

4. Pinouts and pin descriptions

4.1 QFN88L

4.1.1 Pin Diagram

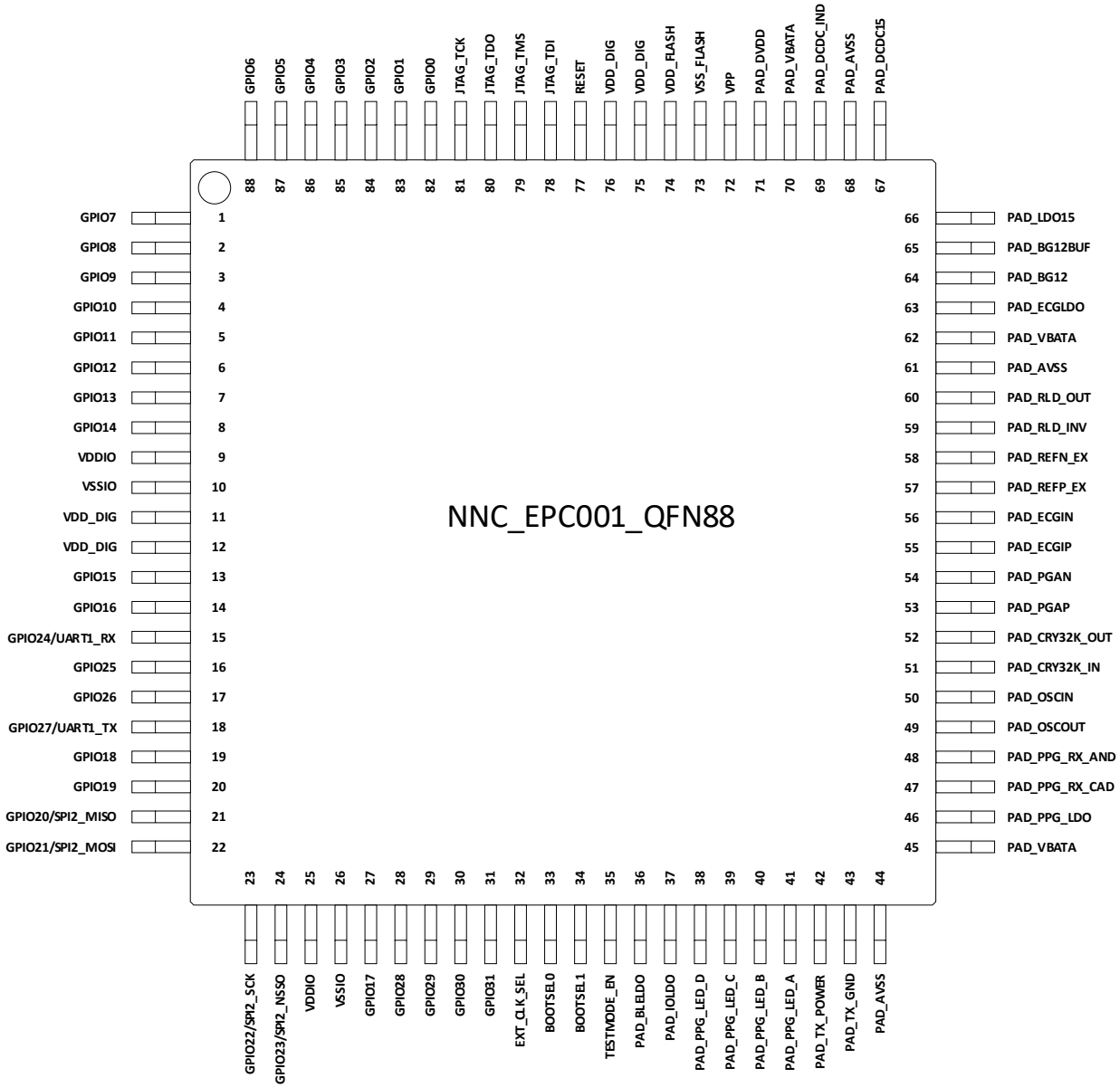


Figure 2 NNC-EPC001 QFN88L Pin information

4.1.2 Pin Description

Pin No.	Pin Name	Pin Type	Pin Primary Function
1	GPIO7	I/O	General Purpose IO7
2	GPIO8	I/O	General Purpose IO8
3	GPIO9	I/O	General Purpose IO9
4	GPIO10	I/O	General Purpose IO10
5	GPIO11	I/O	General Purpose IO11
6	GPIO12	I/O	General Purpose IO12
7	GPIO13	I/O	General Purpose IO13
8	GPIO14	I/O	General Purpose IO14
9	VDDIO	POWER	Digital IO supply
10	VSSIO	GND	Digital IO ground
11	VDD_DIG	POWER	Digital logic power
12	VDD_DIG	POWER	Digital logic power
13	GPIO15	I/O	General Purpose IO15
14	GPIO16	I/O	General Purpose IO16
15	GPIO24	I/O	General Purpose IO24
16	GPIO25	I/O	General Purpose IO25
17	GPIO26	I/O	General Purpose IO26
18	GPIO27	I/O	General Purpose IO27
19	GPIO18	O	BLE – EXT_POR
20	GPIO19	I	BLE – SPI2_INT
21	GPIO20	I/O	BLE – SPI2_MISO
22	GPIO21	I/O	BLE – SPI2_MOSI
23	GPIO22	I/O	BLE – SPI2_SCK
24	GPIO23	I/O	BLE – SPI2_NSS0
25	VDDIO	POWER	Digital IO supply
26	VSSIO	GND	Digital IO ground
27	GPIO17	I/O	General Purpose IO17
28	GPIO28	I/O	General Purpose IO28
29	GPIO29	I/O	General Purpose IO29
30	GPIO30	I/O	General Purpose IO30
31	GPIO31	I/O	General Purpose IO31
32	EXT_CLK_SEL	I	Digital input for external clock select
33	BOOTSEL0	I	Digital input for boot selection bit 0
34	BOOTSEL1	I	Digital input for boot selection bit 1
35	TESTMODE_EN	I	Digital input to select testmode
36	PAD_BLELDO	POWER	LDO output for BLE
37	PAD_IOLDO	POWER	LDO output for IO
38	PAD_PPG_LED_D	ANA_IN	Connect to LED anode

39	PAD_PPG_LED_C	ANA_IN	Connect to LED anode
40	PAD_PPG_LED_B	ANA_IN	Connect to LED anode
41	PAD_PPG_LED_A	ANA_IN	Connect to LED anode
42	PAD_TX_POWER	POWER	Connect to LED to transfer power
43	PAD_TX_GND	GND	Connect to LED to transfer ground
44	PAD_AVSS	GND	PPG ground
45	PAD_VBATA	POWER	PPG power
46	PAD_PPGLDO	POWER	LDO output for PPG
47	PAD_PPG_RX_CAD	ANA_IN	PPG positive input
48	PAD_PPG_RX_AND	ANA_IN	PPG negative input
49	PAD_OSCOUT	ANA_IN	16M Crystal output
50	PAD_OSCIN	ANA_IN	16M Crystal input
51	PAD_CRY32K_IN	ANA_IN	32K Crystal input
52	PAD_CRY32K_OUT	ANA_IN	32K Crystal output
53	PAD_PGAP	ANA_OUT	PGA output
54	PAD_PGAN	ANA_OUT	PGA output
55	PAD_ECGIP	ANA_IN	ECG positive input
56	PAD_ECGIN	ANA_IN	ECG negative input
57	PAD_REFP_EX	ANA_IN	External positive reference
58	PAD_REFN_EX	ANA_IN	External negative reference
59	PAD_RLD_INV	ANA_IN	Right leg drive input
60	PAD_RLD_OUT	ANA_OUT	Right leg drive output
61	PAD_AVSS	GND	ECG ground
62	PAD_VBATA	POWER	ECG power
63	PAD_ECGLDO	POWER	LDO output for ECG
64	PAD_BG12	ANA_OUT	BG Vref output
65	PAD_BG12BUF	ANA_OUT	BG buffer output
66	PAD_LDO15	POWER	1.5v LDO output for digital
67	PAD_DCDC15	POWER	1.5v DCDC output for digital
68	PAD_AVSS	GND	DCDC ground
69	PAD_DCDC_IND	ANA_IN	Connect to external inductor
70	PAD_VBATA	POWER	DCDC power
71	PAD_DVDD	POWER	Digital power (1.5v)
72	VPP	POWER	High voltage for Flash
73	VSS_FLASH	GND	Flash ground
74	VDD_FLASH	POWER	Flash power
75	VDD_DIG	POWER	Digital logic power
76	VDD_DIG	POWER	Digital logic power
77	RESETN	I	External reset pin for digital system
78	JTAG_TDI	I	JTAG interface

79	JTAG_TMS	I	JTAG interface
80	JTAG_TDO	O	JTAG interface
81	JTAG_TCK	I	JTAG interface
82	GPIO0	I/O	General Purpose IO0
83	GPIO1	I/O	General Purpose IO1
84	GPIO2	I/O	General Purpose IO2
85	GPIO3	I/O	General Purpose IO3
86	GPIO4	I/O	General Purpose IO4
87	GPIO5	I/O	General Purpose IO5
88	GPIO6	I/O	General Purpose IO6
89(Thermal pad)	DVSS	GND	Digital ground

Table 4 NNC-EPC001 QFN88L pin list

4.2 QFN68L

4.2.1 Pin Diagram

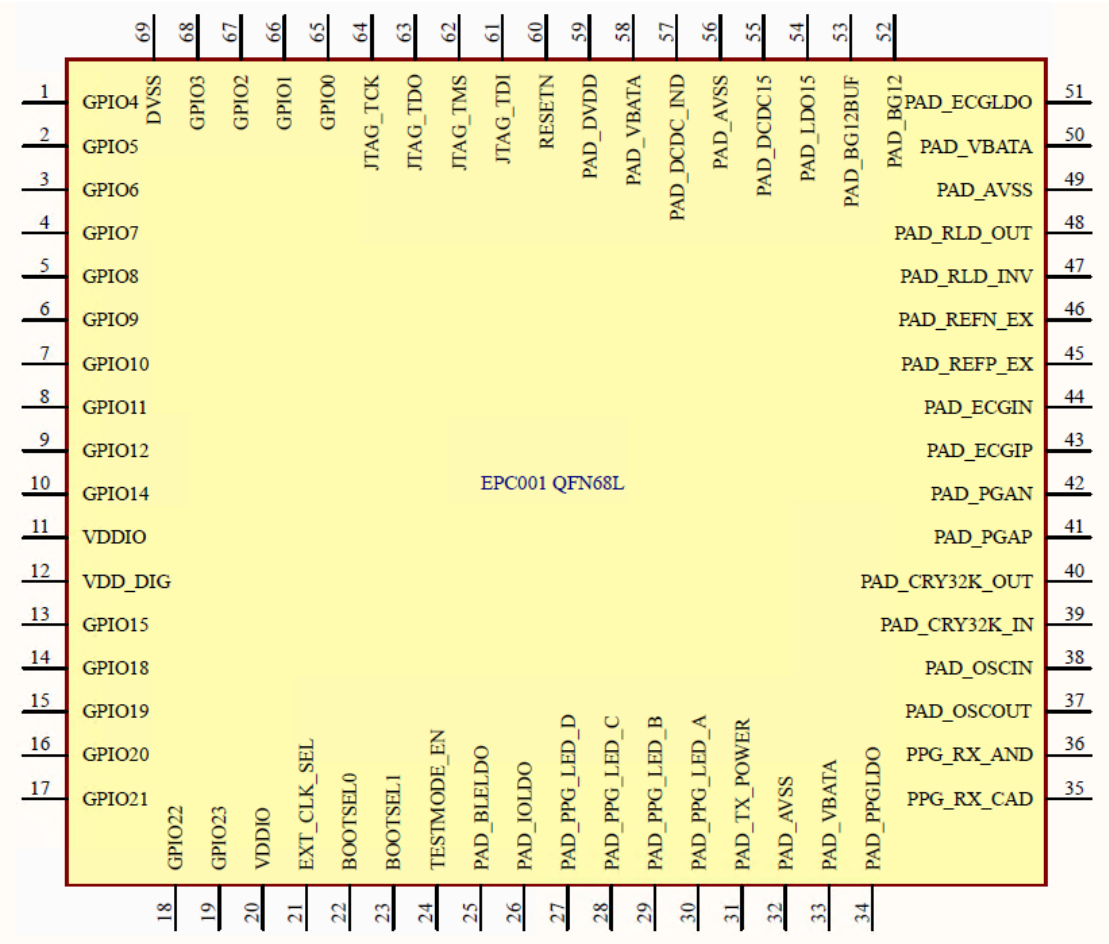


Figure 3 NNC-EPC001 QFN68L Pin information

4.2.2 Pin Description

Pin No.	Pin Name	Pin Type	Pin Primary Function
1	GPIO4	I/O	General Purpose IO4
2	GPIO5	I/O	General Purpose IO5
3	GPIO6	I/O	General Purpose IO6
4	GPIO7	I/O	General Purpose IO7
5	GPIO8	I/O	General Purpose IO8
6	GPIO9	I/O	General Purpose IO9
7	GPIO10	I/O	General Purpose IO10
8	GPIO11	I/O	General Purpose IO11
9	GPIO12	I/O	General Purpose IO12
10	GPIO14	I/O	General Purpose IO14
11	VDDIO	POWER	Digital IO supply
12	VDD_DIG	POWER	Digital logic power
13	GPIO15	I/O	General Purpose IO15
14	GPIO18	O	BLE – EXT_POR
15	GPIO19	I	BLE – SPI2_INT
16	GPIO20	I/O	BLE – SPI2_MISO
17	GPIO21	I/O	BLE – SPI2_MOSI
18	GPIO22	I/O	BLE – SPI2_SCK
19	GPIO23	I/O	BLE – SPI2_NSS0
20	VDDIO	POWER	Digital IO supply
21	EXT_CLK_SEL	I	Digital input for external clock select
22	BOOTSEL0	I	Digital input for boot selection bit 0
23	BOOTSEL1	I	Digital input for boot selection bit 1
24	TESTMODE_EN	I	Digital input to select testmode
25	PAD_BLELDO	POWER	LDO output for BLE
26	PAD_IOLDO	POWER	LDO output for IO
27	PAD_PPG_LED_D	ANA_IN	Connect to LED anode
28	PAD_PPG_LED_C	ANA_IN	Connect to LED anode
29	PAD_PPG_LED_B	ANA_IN	Connect to LED anode
30	PAD_PPG_LED_A	ANA_IN	Connect to LED anode
31	PAD_TX_POWER	POWER	Connect to LED to transfer power
32	PAD_AVSS	GND	PPG ground
33	PAD_VBATA	POWER	PPG power
34	PAD_PPGLDO	POWER	LDO output for PPG
35	PAD_PPG_RX_CAD	ANA_IN	PPG positive input
36	PAD_PPG_RX_AND	ANA_IN	PPG negative input
37	PAD_OSCOUT	ANA_IN	16M Crystal output
38	PAD_OSCIN	ANA_IN	16M Crystal input

39	PAD_CRY32K_IN	ANA_IN	32K Crystal input
40	PAD_CRY32K_OUT	ANA_IN	32K Crystal output
41	PAD_PGAP	ANA_OUT	PGA output
42	PAD_PGAN	ANA_OUT	PGA output
43	PAD_ECGIP	ANA_IN	ECG positive input
44	PAD_ECGIN	ANA_IN	ECG negative input
45	PAD_REFP_EX	ANA_IN	External positive reference
46	PAD_REFN_EX	ANA_IN	External negative reference
47	PAD_RLD_INV	ANA_IN	Right leg drive input
48	PAD_RLD_OUT	ANA_OUT	Right leg drive output
49	PAD_AVSS	GND	ECG ground
50	PAD_VBATA	POWER	ECG power
51	PAD_ECGLDO	POWER	LDO output for ECG
52	PAD_BG12	ANA_OUT	BG Vref output
53	PAD_BG12BUF	ANA_OUT	BG buffer output
54	PAD_LDO15	POWER	1.5v LDO output for digital
55	PAD_DCDC15	POWER	1.5v DCDC output for digital
56	PAD_AVSS	GND	DCDC ground
57	PAD_DCDC_IND	ANA_IN	Connect to external inductor
58	PAD_VBATA	POWER	DCDC power
59	PAD_DVDD	POWER	Digital power (1.5v)
60	RESETN	I	External reset pin for digital system
61	JTAG_TDI	I	JTAG interface
62	JTAG_TMS	I	JTAG interface
63	JTAG_TDO	O	JTAG interface
64	JTAG_TCK	I	JTAG interface
65	GPIO0	I/O	General Purpose IO0
66	GPIO1	I/O	General Purpose IO1
67	GPIO2	I/O	General Purpose IO2
68	GPIO3	I/O	General Purpose IO3
69(Thermal pad)	DVSS	GND	Digital ground

Table 5 NNC-EPC001 QFN68L pin list

5. Functional overview

The NNC-EPC001 SOC is a system on chip embedding one MCU (RISC-V core) and multiple ECG/PPG module suitable for high performance bio electric sensor applications.

Figure 1 shows the general block diagram of NNC_EPC001 system.

5.1 Power management

5.1.1 Voltage Regulator

The NNC_EPC001 is equipped with a linear Low-Dropout regulator (LDO) and a high efficiency step-down DC/DC Buck converter. These blocks provide the digital blocks of the chip with a well-regulated output voltage from a Lithium battery.

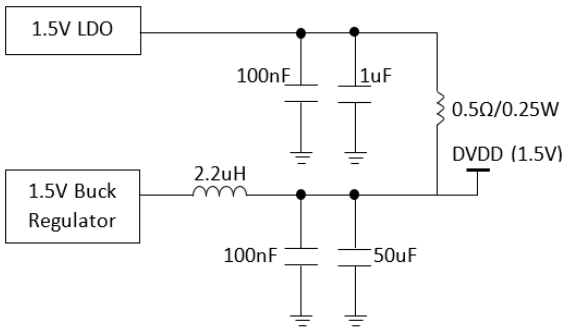


Figure 4 NNC-EPC001 Voltage regulator

During the start-up time by default, the digital blocks are powered up by the LDO generating 1.5V. To save battery energy, user can switch from the LDO to the DC/DC converter using software by configuring the system control block, Ana-LDIO control and Ana-DCDC control registers.

5.1.2 Power Supply schemes

Following are the power supply schemes of NNC_EPC001.

LDO50mA:

The Low Dropout Regulator generates a maximum output current of 50mA. It supplies 3.3V for all IOs.

LDO_PPG:

The Low Dropout Regulator supplies 3.3V for the whole PPG system.

LDO_ECG:

The Low Dropout Regulator supplies 3/3.3/3.6/3.9/4.2V for the whole ECG system.

BANDGAP_BUFFER:

The Low noise Bandgap outputs 1.2V/2.4V reference voltage to ECG system.

LDO_BLE:

The Low Dropout Regulator supplies 3.3V.

DC/DC Buck Converter:

The high efficiency DC/DC buck converter generates 1.5V output voltage for the digital circuit.

LDO15:

The Low Dropout Regulator supplies 1.5V for the digital circuit.

The figure illustrates the power supply distribution for NNC_EPC001 chip. Please refer 'Application example component list' section 7.1 for the recommended external parts and connections.

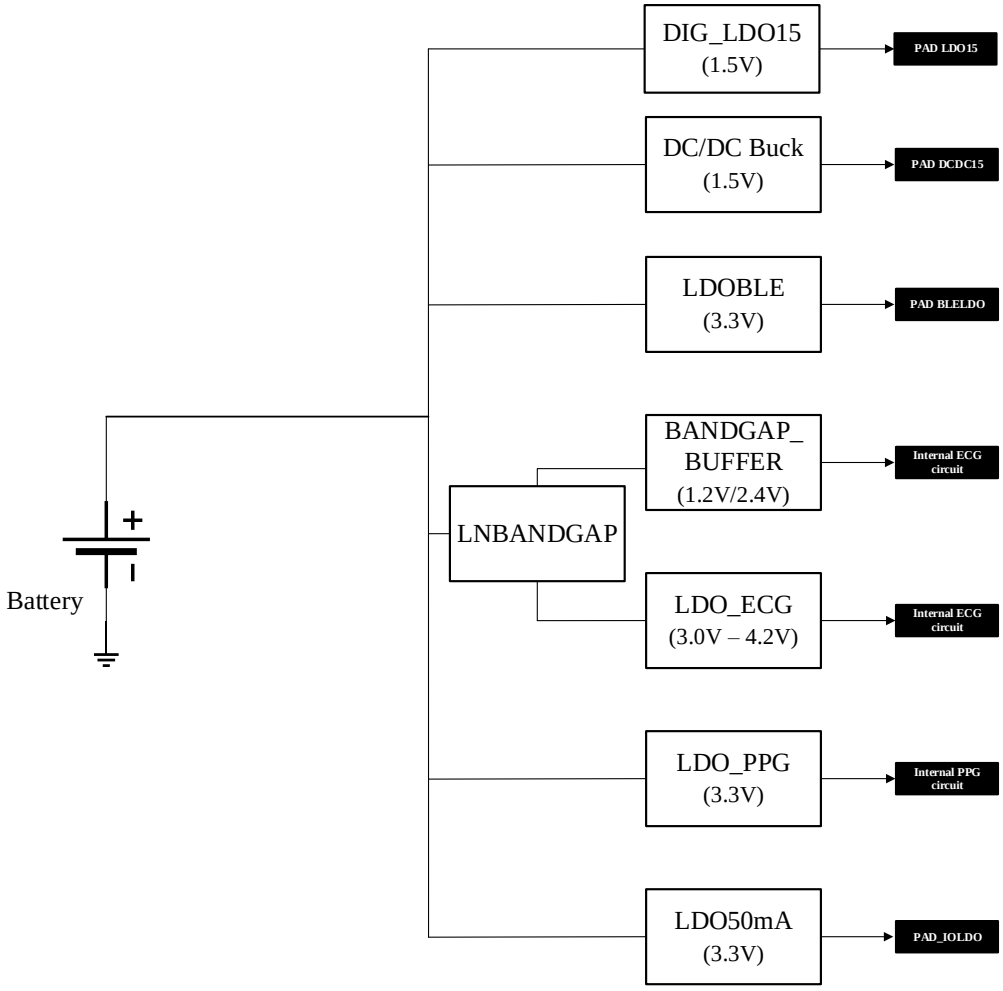


Figure 5 NNC-EPC001 Power distribution

5.1.3 Low-Power modes

The NNC_EPC001 supports two low-power modes as shown:

- Sleep mode:

When CPU issues WFI instruction, the system enters sleep mode. When system is in sleep mode, the clocks of main units inside core will be gated off. The state transition events from sleep to active mode include all interrupts.

- Stop mode:

The stop mode can be activated by enabling GO_STOP_EN of PMU control register prior to issuing WFI instruction. In this mode all high frequency clocks will be gated off. But the low frequency clock will be always on. Therefore, this mode is also called Power Saving mode. The state transition events from stop to active mode only include:

  - RTC alarm event
  - RTC periodic wakeup event
  - GPIO interrupts

5.1.4 Active mode

Upon power on reset, the NNC_EPC001 will be working in normal active mode, in which all the clocks will be running normally. The power management digital control unit manages the transition from Active mode to Stop(power saving) mode and vice versa.

The figure below illustrates the transition between Active mode and Stop mode.

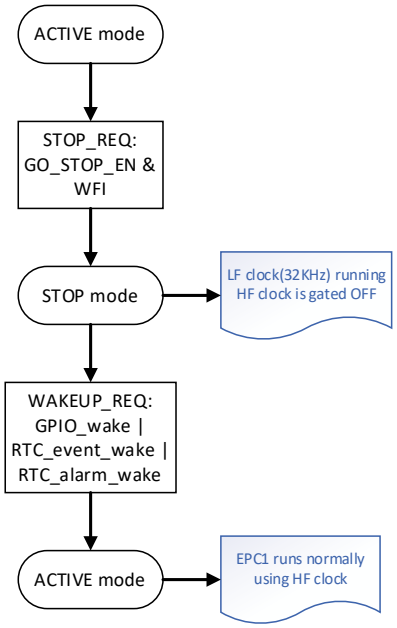


Figure 6 NNC-EPC001 Power-saving mode transition

5.2 MCU core

The SiFive S2 Standard Core is a high-performance, full-featured embedded processor designed to address advanced microcontroller applications. It is highly optimized for area and power while still offering class-leading performance. The S2 includes a 32-bit S2 RISC-V core, which has an efficient, single-issue, in-order execution pipeline, with a peak execution rate of one instruction per clock cycle. The S2 core supports machine and user privilege modes, as well as standard Multiply (M) and Compressed (C) RISC-V extensions (RV32IMC).

It has the following main features:

- Harvard Architecture with separate CPU instruction fetch and data load/store interfaces
- Integrated 8Kbyte direct mapped uCache for instructions
- Core Local Interrupt Controller (CLIC) to support 64 Level Triggered Interrupts
- 32Kbyte + 32Kbyte Tightly Integrated SRAM Memories
- Physical Memory Protection
- Optimized crossbar interconnect between CPU, TIM, peripherals and external AHB-lite ports which are
    - 2 System AHB-lite Master ports
    - 1 Peripheral AHB-lite Master port
    - 1 Front AHB-lite Slave port
- A debug unit to support 4 wire JTAG based debugger host connection

Below figure shows the block diagram of S2 RISC-V CPU core complex

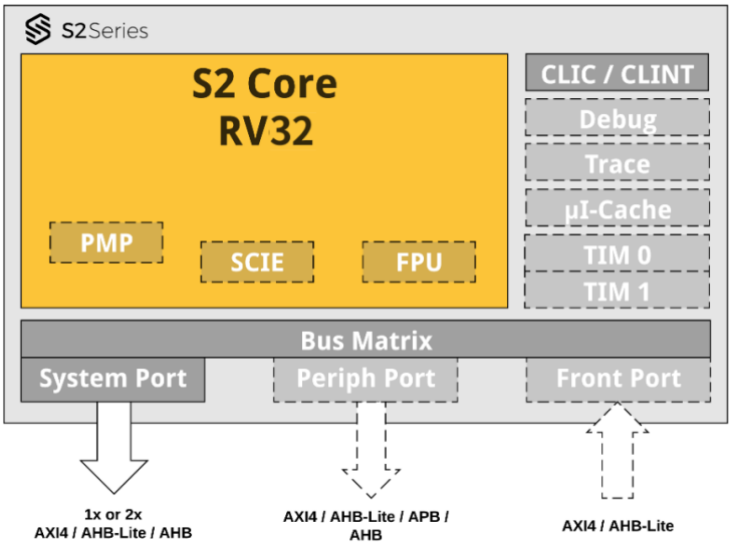


Figure 7 NNC-EPC001 MCU riscv core

5.3 Memories

5.3.1 FLASH

The NNC_EPC001 includes an embedded Flash memory IP, with x32 configuration, Sector Erase, Chip Erase, and Byte Program capability. It is organized as 64K words/256K bytes main array with Eight Non-Volatile Register sectors (NVR). Each sector has 128 words/512 Bytes.

All sectors are readable/writeable except for NVR Sector 8 which is only readable. Also, the NVR Sector 8 is protected from erase and contains configuration data required to set the configuration registers. The NVR Sectors can only be erased via Sector Erase. Chip Erase will erase the entire Main Array.

The Flash controller supports the following functions:

- The Flash IP can be placed in DEEP SLEEP mode with significantly lowered leakage current in order to save power.
- Unprogrammed destination word check before programming.
- The Flash controller also supports generation of interrupt from two events namely End of Operation and Programming Error.
- The AHB Bus Stalling feature with one outstanding transaction while controller is busy
- There is additional write protection for NVR sectors from 1-8
- Built-in Self Test (BIST) with serial input or output through pads

9.3.1.1 Timing Parameters

The NNC_EPC001 Flash timing parameters are programmable to support operation across a varying frequency range. These values are calculated in terms of number of clocks based on the operating AHB clock. They have to be initialized in the Timing registers 0~7 before performing any program or erase.

The table below shows the timing parameters of NNC_EPC001 FLASH.

Details	Parameter	Unit	Min	Max
Read Cycle Time	Read_access	ns	25	25
Wake up time from deep Standby to any operation	Twup	us	3.5	
Duration of PORb High to any operation	Trhr	us	3.5	
Address/Din/CEb to CONFEN High Setup time	Tcfs	ns	40	
Address/Din/CEb to CONFEN Low Hold time	Tcfh	ns	40	
CONFEN pulse width	Tconfen	ns	40	
PROG/ERASE/CEb/NVR/Address to WEb Setup Time	Tnvs	us	6	
Byte Program Time	Tprog	us	6	7.5
BYTE<3:0>/Address/Data Setup Time	Tads	ns	15	
BYTE<3:0>/Address/Data Hold Time	Tadh	ns	15	
WEb low to PROG2 high Setup Time	Tpgs	us	5	6.5
PROG2 Low to WEb high hold time	Tpgh	ns	15	
Program Recovery time	prog_Trcv	us	5	
Sector Erase Recovery time	ser_Trcv	us	50	
Chip Erase Recovery time	mer_Trcv	us	200	
Sector Erase Duration	ser_Terase	ms	4	5
Chip Erase Duration	mer_Terase_chip	ms	30	40
Latency to next operation after PROG/ERASE low	Trw	ns	100	

Table 6 NNC-EPC001 Flash Timing parameters

9.3.1.2 User Configuration

1. Word Program
   a. Set PG bit<0> of control register
   b. Issue word write via direct access interface
   c. Wait for EOP in status register and Clear EOP
2. Sector Erase
   a. Set SER bit<1> and SECT_ADD in control register
   b. Start Erase by writing 1 to STRT bit<16> of control register
   c. Wait for EOP in status register and Clear EOP
3. Mass/Chip Erase
   a. Set MER bit<2> of control register
   b. Start Erase by writing 1 to STRT bit<16> of control register
   c. Wait for EOP in status register and Clear EOP

5.3.2 SRAM

In NNC_EPC001 core complex, the SRAM is used for instruction cache data memory, instruction cache tag memory, Tightly-Integrated memory (TIM 0) & Tightly-Integrated memory (TIM 1). The NNC_EPC001 core complex has 64 KB of Tightly-Integrated Memory (TIM) split evenly over two contiguous banks (TIM 0 and TIM 1). Having split TIMs allows for simultaneous access to both banks. When executing code solely from TIM address space, for maximum performance, it is recommended to place code in one TIM and data in the other. If executing code from an off core complex memory device, such as an AHB flash controller, the TIMs can be treated as a single contiguous address space.

Each of the 32KB TIMs are implemented with 2X16K SRAM block.

The NNC_EPC001 also includes SRAM BIST controller that is based on March C+ algorithm, to support BIST testing through pads.

5.3.3 ROM

The NNC-EPC001 includes 17 ROM IPs. Out of these, 8 ROM blocks whose size is 2048x32bits (i.e., 8Kb) are used as AHB ROM. The remaining 8 ROM blocks whose size is 128x24bits, are used in ECG low pass filter and one other ROM IP with size 256x24bits is used in ECG band stop filter.

The rom controller connected to the peripheral port controls the access of 64Kb AHB ROM (split as eight 8Kb ROM IPs). It also includes ROM BIST controller to perform a readout of all the contents of the ROMs.

5.4 Boot modes

The boot selection pins decide the Reset Vector of CPU, which is the address at which CPU boot upon reset.

Four boot modes can be selected through the BOOTSEL0 and BOOTSEL1 pins, as shown in the following table.

Boot mode selection		Reset Vector (Address)	Boot mode
BOOTSEL1 pin	BOOTSEL0 pin		
0	0	Flash NVR (0x20140000)	Flash NVR region is selected as boot area for bootloader usage
0	1	Flash Main Array (0x20100000)	Flash main memory is selected as boot area for application code usage
1	0	SRAM (0x80000000)	Embedded SRAM (TIM0) is selected as boot area for debug usage
1	1	ROM (0x20000000)	Boot from ROM is reserved for future use. Not supported in current NNC_EPC001 version

Table 7 NNC-EPC001 Boot selection

5.5 Clock and start-up

5.5.1 Clock Structure

The NNC_EPC001 has two main clock domains namely the low-frequency (LFCLK) and high-frequency (HFCLK) domain.

The main clocks in these domains are mentioned here:

- HFCLK_RC16M:
  It is a 16MHz high-frequency clock generated by a fully-integrated RC Oscillator. After power on reset, this high-speed RC clock will be selected as default HFCLK.
- HFCLK_XTAL16M:
  It is a 16MHz high-frequency clock generated by a Crystal Oscillator. In order to set this clock as HFCLK, user need to reset the CLK16M_SEL bit<1> of PMU system control register1.
- HFCLK_EXT:
  It is a high-speed external clock, which can be up to 16MHz. It can be driven using GPIO0 when EXT_CLK_SEL pin is pulled to high.
- HFCLK_PLL:
  It is a high-speed clock generated by PLL, which can be up to 64MHz. In order to set this clock as HFCLK, user should enable PLL by configuring ANA-PLL system control registers. Also, user need to set the INT_HFCLK_SEL bit<3> of PMU system control register1.
- LFCLK_RC32K:
  It is a 32KHz low-frequency clock generated by a fully-integrated RC oscillator. After power on reset, this low-speed RC clock will be selected as default LFCLK.
- LFCLK_XTAL32K:
  It is generated by low-speed oscillator with external crystal resonator. It generates an accurate frequency of 32.768KHz. In order to set this clock as LFCLK, user need to reset the LFCLK_SEL bit<2> of PMU system control register1.

The system clock selection is performed on startup, however the internal RC 16MHz oscillator (HFCLK_RC16M) is selected as default CPU clock on reset. Several prescalers allow the application to configure the frequency of the AHB and APB clock domains.

5.5.2 Reset Sources

The NNC_EPC001 system reset is asynchronous reset, where the reset put the system in to the known state without depending on the clock. So, the reset controller of NNC_EPC001 uses a reset synchronizer for each clock domains. It also uses an ordered proper sequence of reset removal method.

The reset sources are:

- External reset
- Power on reset
- Internal reset requests from different design domains are:
  - system software reset request
  - watchdog reset request
  - software reset request for APB peripherals

5.6 General purpose inputs/outputs (GPIOs)

The NNC_EPC001 supports 32 Multifunction GPIOs and 9 Dedicated digital IO's driven through 41 IO Buffers. Most of the GPIO pins can support alternate and analog functions as shown in following table.

5.6.1 GPIO Alternate Functions

Pin No.	GPIO Name	Alternate Function 0	Alternate Function 1	Alternate Function 2	Alternate Function 3	Analog Function
82	GPIO0	I2C0_SDA	-	SPI1_SCK	HFCLK_EXT	GPIO_AN_IP
83	GPIO1	I2C0_SCL	-	SPI1_MOSI	-	GPIO_AN_IN
84	GPIO2	TIMER0_EXTIN	-	SPI1_MISO	-	AN3
85	GPIO3	TIMER1_EXTIN	-	SPI1_NSS0	-	AN2
86	GPIO4	I2S0_CLK	UART0_RXD	SPI1_NSS1	QSPI_SCK	DCDC_TESTPIN
87	GPIO5	I2S0_WS	UART0_CTS_N	SPI1_NSS2	QSPI_BK1_NCS	DCDC_CTRL
88	GPIO6	I2S0_DAT	UART0_RTS_N	SPI1_NSS3	QSPI_BK1_IO0	PPG_RX_VOUTN
1	GPIO7	-	UART0_TXD	-	QSPI_BK1_IO1	PPG_RX_VOUTP
2	GPIO8	-	-	-	QSPI_BK1_IO2	MON_VTUNE
3	GPIO9	SPI0_SCK	-	-	QSPI_BK1_IO3	TEST_IB50NA
4	GPIO10	SPI0_MOSI	PWM1	-	QSPI_BK2_NCS	TEST_IB1U
5	GPIO11	SPI0_MISO	PWM2	-	QSPI_BK2_IO0	-
6	GPIO12	SPI0_NSS0	PWM3	-	QSPI_BK2_IO1	-
7	GPIO13	SPI0_NSS1	-	PWM4	QSPI_BK2_IO2	-
8	GPIO14	SPI0_NSS2	I2C1_SDA	PWM5	QSPI_BK2_IO3	-
13	GPIO15	SPI0_NSS3	I2C1_SCL	PWM6	-	-
14	GPIO16	-	-	-	-	-
27	GPIO17	-	-	-	-	-
19	GPIO18	-	-	-	-	-
20	GPIO19	-	-	-	-	-
21	GPIO20	-	-	BLE-SPI2_MISO	-	-
22	GPIO21	-	-	BLE-SPI2_MOSI	-	-
23	GPIO22	-	-	BLE-SPI2_SCK	-	-
24	GPIO23	-	-	BLE-SPI2_NSS0	-	-
15	GPIO24	-	-	SPI2_NSS1	UART1_RXD	-
16	GPIO25	-	-	SPI2_NSS2	UART1_CTS_N	-
17	GPIO26	-	-	SPI2_NSS3	UART1_RTS_N	-
18	GPIO27	-	-	-	UART1_TXD	-
28	GPIO28	-	I2S1_CLK	-	-	-
29	GPIO29	-	I2S1_WS	-	-	-
30	GPIO30	-	I2S1_DAT	-	-	-
31	GPIO31	-	-	-	-	-

Table 8 NNC-EPC001 GPIO alternate functions

The 9 Dedicated IOs are explained below:

RESETN:

RESETN is an active low external reset pin, which will reset the entire chip when asserted.

TESTMODE_EN:

The testmode functions for the IOs are activated using TESTMODE_EN. When testmode is enabled, the BOOTSEL1 and BOOTSEL0 pins are used to select different testmode functions.

JTAG pins:

JTAG_TCK, JTAG_TDO, JTAG_TMS and JTAG_TDI are the four dedicated pins for JTAG interface.

EXT_CLK_SEL:

EXT_CLK_SEL is dedicated to select external clock. It will enable driving an external high frequency clock through GPIO0.

BOOTSEL <1:0>:

BOOTSEL1 and BOOTSEL0 pins are dedicated to select boot mode. This controls the Reset Vector of the CPU.

5.6.2 GPIO Modes

Each of the 32 GPIO pins can be configured in different modes by setting the MODE bits in corresponding GPIO Register as follows:

- 00 – Input Mode
- 01 – Output Mode
- 10 – Alternate function Mode
- 11 – Analog Mode

In order to set the function mode these 2 Mode bits are assigned for each GPIO. Therefore, a total of 64-bits spanned across the two 32-bit GPIOL16_MODE and GPIOH16_MODE registers respectively. GPIOL16_MODE specifies the Mode bits for GPIO0 through 15 and GPIOH16_MODE specifies the Mode bits for GPIO16 through 31.

5.6.3 User Configuration

1. Input mode
    a. Set mode bits 00 in the corresponding GPIO MODE register
    b. When configured in this mode, output buffers are disabled to allow values from the PAD to be driven into the input buffers and sampled by the GPIO Datain registers.
2. Output mode
    a. Set mode bits 01 in the corresponding GPIO MODE register
    b. When configured in this mode, both input and output buffers are enabled. The PAD is driven by the output buffer through GPIO Dataout Registers and this is driven back via the input buffer to the GPIO Datain Registers.
3. Alternate function mode
    a. Set mode bits 10 in the corresponding GPIO MODE register
    b. GPIOL16_ALTFCN specifies the Alternate Function bits for GPIO0 through 15 and GPIOH16_ALTFCN specifies the Alternate Function bits for GPIO16 through 31. Set the 2 bits alternate function for the GPIO as follows.

i. 00 – Alternate Function 0
ii. 01 – Alternate Function 1
iii. 10 – Alternate Function 2
iv. 11 – Alternate Function 3

c. This alternate function setting allows up to 4 different peripheral inputs/outputs to be muxed into one particular IO.

4. Analog mode
   a. Set mode bits 11 in the corresponding GPIO MODE register
   b. When configured in this mode, both input and output buffers are disabled to allow the analog signal source/sink to be driven to/from the PAD.

5.7 Direct memory access controller (DMAC)

The DMA controller is compliant with AMBA2.0 AHB-lite Specification. It manages memory-to-memory, memory-to-peripheral and peripheral-to-memory transfers. It has two interfaces; slave interface as well as master interface. The NNC_EPC001 DMAC slave interface is connected to the System 1 port and the master interface is connected to the Front port.

5.7.1 Trigger Sources

The NNC_EPC001 DMA controller supports maximum 16 channels. Each channel is connected to dedicated hardware DMA triggers, with support for software trigger as well. The configuration is made by software to set the source address, destination address and transfer data size.

The 16 DMA Channels is wired to 16 triggering sources as listed below.

DMA Channel	Trigger Source
15	ECG
14	PPG
13	QSPI
12	I2C1
11	I2C0
10	I2S0 / I2S1
9	SPI2 RX
8	SPI2 TX
7	SPI1 RX
6	SPI1 TX
5	SPI0 RX
4	SPI0 TX
3	UART1 RX
2	UART1 TX
1	UART0 RX
0	UART0 TX

Table 9 NNC-EPC001 DMA sources

5.7.2 User Configuration

1. Write source address in Channel n Source Address register (SARn)
2. Write destination address in Channel n Destination Address register (DARn)
3. Configure BLOCK_TL in Channel n Control register A (CHn_CTRL_A), such that BLOCK_TL = Block transfer size (in bytes) – 1
4. The DMA transfer mode can be selected by configuring TRGTMDC (Trigger transfer mode control) of Channel n Control register B (CHn_CTRL_B)
5. If Single Transfer mode, only single data based on hsize will be transferred:
    a. Set TRGTMDC as 00
    b. Configure GRPMC field of CHn_CTRL_A register, which indicates whether the single data transfer is from Source or Destination
6. If Block Transfer mode, the entire block based on block transfer size will be transferred at one time:
    a. Set TRGTMDC as 10
7. If Group Transfer mode, the entire block based on block transfer size will be transferred as several groups:
    a. Set TRGTMDC as 01
    b. Set the GROUP_LEN = Group length - 1. Group length is the No: of bytes to be transferred for each group.
    c. Set SGRPADDRC = 1, if the Source Group Address should start at the same location (as stated in SARn) for each group transaction
    d. Set DGRPADDRC = 1, if the Destination Group Address should start at the same location (as stated in DARn) for each group transaction
8. Configure SINC field of CHn_CTRL_A register as below:
    a. Set SINC = 00, which increments the source address on every transfer
    b. Set SINC = 01, which decrements the source address on every transfer
    c. Set SINC = 1X, which will use the same source address on every transfer
9. Configure DINC field of CHn_CTRL_A register as below:
    a. Set DINC = 00, which increments the destination address on every transfer
    b. Set DINC = 01, which decrements the destination address on every transfer
    c. Set DINC = 1X, which will use the same destination address on every transfer
10. Set up the Source transfer width and Destination transfer width based on hsize in SRC_TR_WIDTH and DST_TR_WIDTH of CHn_CTRL_A register respectively
11. Channel Protection control can be enabled using PROTCTL of CHn_CTRL_B register
12. Source read data Endian control can be set using SRCDTLGC of CHn_CTRL_B register
13. Destination write data Endian control can be set using DSTDTLGC of CHn_CTRL_B register
14. If needed all DMA Interrupts can be enabled using INT_EN of CHn_CTRL_B register
15. Finally, user can enable the Global DMA control bit of DMAC Channel Configuration register (DMACCFG) and then set the corresponding channel Enable in Channel n Enable control register (CHn_EN)

5.8 Interrupts and events

There are 64 interrupt lines supported by the CPU core. All incoming interrupts are synced to the CCXCLK (core complex clock) domain.

IRQ	Function	Description
0	Timer 0	Timer 0 timeout interrupt
1	Timer 1	Timer 1 timeout interrupt
2	Dual Timer	Dual Timer timeout interrupt
3	I2S 0	I2S0 Data buffer/ Overrun/ Underrun status interrupt
4	I2S 1	I2S1 Data buffer/ Overrun/ Underrun status interrupt
5	UART 0	UART0 Data buffer/ Line/ Modem status interrupt
6	UART 1	UART1 Data buffer/ Line/ Modem status interrupt
7	I2C 0 Event	I2C0 Data buffer/ Transfer Event status interrupt
8	I2C 1 Event	I2C1 Data buffer/ Transfer Event status interrupt
9	Watchdog	Watchdog Timer timeout interrupt
10	SPI 0	SPI0 Data buffer/ Transfer complete/ Overrun/ Underrun status interrupt
11	SPI 1	SPI1 Data buffer/ Transfer complete/ Overrun/ Underrun status interrupt
12	SPI 2	SPI2 Data buffer/ Transfer complete/ Overrun/ Underrun status interrupt
13	RTC	RTC Alarm interrupt/ Period Wakeup Timer interrupt
14	I2C 0 Error	I2C0 Error status interrupt
15	I2C 1 Error	I2C1 Error status interrupt
16	DMA Ctrl	DMA Transfer status interrupt
17	Quad SPI	Quad SPI Data buffer/ Transfer complete/ Timeout/ Polling status match/ Error status interrupt
18	Flash Ctrl	End of operation/ Program Error status interrupt
19	PWM	PWM match status interrupt
20	HEARTAPP	Measurement End status interrupt
21	PPG	PPG Data buffer status interrupt
22	ECG	ECG Data buffer/ ADC EOC/ LP, HP, BS filter Calculation end/ Leadoff Detection/ Overrun/ Underrun status interrupt
23	System ctrl LVD	LVD out status interrupt
24	System ctrl PLL	PLL status interrupt
25	System ctrl BG	BG status interrupt
26-31	-	-
32-63	GPIO32	GPIO 0-31 individual interrupts

Table 10 NNC-EPC001 Interrupt sources

5.9 Timers and Watchdogs

The NNC_EPC001 has two general purpose timers, a dual-timer and a PWM. It also has an independent watchdog timer.

5.9.1 General-purpose timers

The NNC_EPC001 has two 32-bit general-purpose high frequency down-counters, namely TIMER0 and TIMER1. It can generate interrupt when counter reaches 0. By, default the timers work on APB clock. It also supports counting based on an external pin which can serve as an enable or a clock.

9.9.1.1 User Configuration

1. Timer based on internal APB clock
    a. Set Timer interrupt enable, INT_EN in control register
    b. Config Timer counter CURRENT value & RELOAD value
    c. Start timer using timer enable, TMR_EN in control register
2. Timer based on External clock (GPIO)
    a. To drive clock in external pin of TIMER0, set GPIO2 in Alternate function 0 mode. For TIMER1, set GPIO3 in Alternate function 0 mode
    b. Set Timer interrupt enable, INT_EN in control register
    c. Config Timer counter CURRENT value & RELOAD value
    d. Set EXT_CLK and TMR_EN in control register
3. Timer based on External enable (GPIO)
    a. To drive enable in external pin of TIMER0, set GPIO2 in Alternate function 0 mode. For TIMER1, set GPIO3 in Alternate function 0 mode
    b. Set Timer interrupt enable, INT_EN in control register
    c. Config Timer counter CURRENT value & RELOAD value
    d. Set EXT_EN and TMR_EN in control register

5.9.2 Dual-Timer

The dual timer is an APB dual-input timer module consisting of two programmable 32-bit or 16-bit down-counters that can generate interrupts when they reach zero. The operation of each timer module is identical. They operate in three modes which are Free-Running mode, One-shot count mode and Periodic mode.

9.9.2.1 User Configuration

1. Select the 16-bit or 32-bit timer operation using TIMER_SIZE in control register.
2. Select the mode of operation by configuring ONE_SHOT and TIMER_MODE bits.
3. Enable timer interrupt using INT_EN in control register.
4. Load the counter value to the TIMn_LOAD bits.
5. Enable the timer by setting TIMER_EN in control register.

5.9.3 PWM

The PWM is based on a 32-bit Timer counter and inherits all of its features. The PWM function in addition to these features, is based on match register events. The ability to separately control rising and falling edge locations allows the PWM to be used for more applications. The NNC_EPC001 has 6 PWM output channels. There are seven match registers which allows up to 6 single edge-controlled or 3 double edge-controlled PWM outputs, or a mix of both types.

9.9.3.1 User Configuration

1. Write proper GPIO alternate registers: configure GPIO for the 6 PWM outputs
2. Enable PWM channels by setting PWMEN of corresponding channel in the PWM control register (PWM_PCR)
3. By default, each channel operates in single edge mode. Can switch to double edge mode by setting PWMSEL in PWM_PCR.
4. Now Load the match register values in PWM Match registers 0~6, which will be loaded in the counters
5. Configure PWM Match Control register (PWM_MCR) to control the pulse output generated in each channel
6. Set the 32-bit Prescaler value in PWM_PR register
7. Enable bits in Load Enable register (PWM_LER), which will activate automatic reload of counter from match register values
8. Finally enable PWM Timer by setting CNT_EN = 1 in PWM Timer control register (PWM_TCR)
9. If needed, the interrupts can be enabled by setting appropriate bits in PWM_MCR

5.9.4 Watchdog timer

The primary function of the watchdog timer (WDT) module is to perform a controlled system restart after a software problem occurs. The time-out period is user configurable and when the selected time interval expires, a system reset is generated. If the watchdog function is not needed in an application, the module can be configured as an interval timer and can generate interrupts at selected time intervals.
The watchdog interrupt in NNC_EPC001 is a Non-Maskable Interrupt (NMI) with higher priority. Also, the watchdog register interface has a software-controlled lock to prevent accidental write access in the watchdog registers.

9.9.4.1 User Configuration

4. Loading the Watchdog Counter
   d. Unlock watchdog by writing 0x1ACCE551 to Watchdog Lock register
   e. Load the watchdog count by writing to Watchdog Load register
   f. Lock the watchdog by writing 0 to the Watchdog Lock register
5. Setting Watchdog for Non Maskable Interrupt (NMI) generation
   Enable interrupt in Watchdog Control register to generate NMI after configured count
6. Setting Watchdog for RESET generation
   a. Enable reset in Watchdog Control register
   b. Check System Control PMU Status reset information bit, WDRESETREQ

7. Issuing manual NMI using the Integration Test Register
   a. Write 1 into the ITCR register
   b. Write 2 into ITOP register
8. Issuing manual Watchdog Reset using the Integration Test Register
   a. Write 1 into the ITCR register
   b. Write 1 into ITOP register

5.10 Real-Time Clock (RTC)

The Real-Time Clock (RTC) provides an automatic wakeup to manage low-power mode. The RTC is an independent BCD timer/counter. Its main function is to keep track of the current time.

The NNC_EPC001 RTC has following features:

- Calendar with seconds, minutes, hours (12 or 24 format), week day, date, month and year
- Compensations for 28-, 29-(leap year), 30-, and 31-day months are performed
- Binary or BCD representation of time, calendar and alarm
- Two interrupts are separately software maskable
  - time-of-day clock/calendar with programmable alarm interrupt
  - periodic programmable counter wakeup interrupt

5.10.1 User Configuration

1. Calendar Initialization
   a. To generate a 1Hz clock for the calendar counter, program prescaler.
   b. Set INIT bit to 1 and configure DATA_MODE and HOUR_MODE in the RTC_CR register to enter initialization mode.
   c. Load the initial time and date values in the RTC_TR and RTC_DR registers.
   d. Exit the initialization mode by clearing the INIT bit.
   e. Note: When INIT_SYNC_READY bit in RTC_SR register is 0, software cannot set INIT bit to trigger another initialization. Also, to read the calendar after initialization, software must check whether the INIT_SYNC_READY bit is 1.

2. Programming the alarm
   a. Follow the steps in calendar initialization (1), but also program the time alarm RTC_TAR and date alarm RTC_DAR registers.
   b. Set ALARM_EN in RTC_CR register to enable alarm.
   c. In order to generate interrupt, Set ALARM_IE in RTC_IER register.

3. Programming the Periodic Wakeup Timer
   a. Program the periodic wakeup input clock-select in RTC_CR register.
   b. Program the periodic wakeup clock prescaler in RTC_WPR register.
   c. Program the periodic wakeup timer value in RTC_WTR register.
   d. Set WUT_EN in RTC_CR register to enable the wakeup timer.
   e. In order to generate interrupt, Set WUT_IE in RTC_IER register.

5.11 Inter-integrated circuit interface (I^2C)

The NNC_EPC001 has two I^2C interfaces (I2C0 & I2C1) which can operate in multi-master or slave modes. Both can support Standard mode (up to 100kbit/s), Fast mode (up to 400kbit/s) and High-Speed mode (up to 3.4Mbit/s). Both support 7-bit and 10-bit addressing modes. They also support data transfer using DMA.

The I2C interface can operate in the following modes for either transmission or reception:

- I2C Slave mode
- I2C Master mode

By default, the I2C interface operates in Slave mode. To switch from default Slave mode to Master mode a Start condition generation is needed. In Master mode, the I2C interface initiates a data transfer and generates the clock signal.

5.11.1 User Configuration

Write proper GPIO alternate registers: configure GPIO for SDA and SCL pins.

1. I2C Slave mode
    a. Configure interrupt in Control_2 register (I2C_CR2)
    b. Configure PE of Control_1 register (I2C_CR1) to enable I2C interface
    c. Configure Own address register (I2C_OAR) to set I2C slave address
    d. Configure ACK of I2C_CR1 to set ack bit
    e. Wait for address match interrupt and clear ADDR bit of status register (I2C_SR)
    f. If acting as a Transmitter, write the transmit data to I2C_DR register
    g. If acting as a Receiver, wait until RXNE of I2C_SR is set, then read the received data from I2C_DR register
2. I2C Master mode
    a. Configure interrupt and clock frequency in I2C_CR2 register
    b. Configure PE of I2C_CR1 to enable I2C interface
    c. Configure START of I2C_CR1 to start I2C master
    d. Wait till start bit is generated by master by reading the SB of I2C_SR register.
    e. Follow further steps below if 7-bit addressing mode:
        i. Write slave address to I2C_DR
        ii. Wait for address sending by reading ADDR bit of I2C_SR
        iii. If acting as a transmitter, write the transmit data to I2C_DR
        iv. If acting as receiver, Wait for RXNE to read the received data from I2C_DR. Also acknowledge by setting ACK=0 in I2C_CR1.
    f. Follow further steps below if 10-bit addressing mode:
        i. Write 10-bit address header to I2C_DR
        ii. Wait for address header sending by reading ADD10 bit of I2C_SR. Then write the slave address to I2C_DR.

iii. If acting as a transmitter, wait for address sending by reading ADDR bit of I2C_SR. And then write the transmit data to I2C_DR

iv. If acting as receiver, wait for address sending by reading ADDR bit of I2C_SR. And then wait for RXNE to read the received data from I2C_DR. Also acknowledge by setting ACK=0 in I2C_CR1.

g. Configure STOP of I2C_CR1 to stop I2C master

NOTE: By setting DMAEN in I2C control 2 register, DMA transfer can be activated.

5.12 Universal asynchronous receiver/transmitter (UART)

The NNC_EPC001 embeds two universal asynchronous receivers/transmitters (UART0 and UART1). They perform serial-to-parallel conversions on data received from a peripheral device and parallel-to-serial conversion on data received from the CPU. The NNC_EPC001 APB UART on power up will be in single character mode, but it can also operate in FIFO mode. They can support up to 115200bps baud rate.
This fully programmable serial interface has the following data characteristics:

- Data can be 5, 6, 7, 8 bits
- Even, odd or no-parity bit generation and detection
- 1, 1.5 or 2-stop bit generation

The NNC_EPC001 UART also supports an autoflow control mechanism which will eliminate overrun errors.
The UART interfaces can be served by the DMA controller.

5.12.1 User Configuration

1. Perform the necessary device pin multiplexing setup by configuring GPIO alternate registers for TXD/RXD
2. Set the desired baud rate by writing the appropriate clock divisor values to the divisor latch registers (DLL and DLH).
3. If the FIFOs will be used, select the desired trigger level and enable the FIFOs by writing the appropriate values to the FIFO control register (FCR). The FIFOEN bit in FCR must be set first, before the other bits in FCR are configured.
4. Choose the desired protocol settings by writing the appropriate values to the line control register (LCR).
5. If autoflow control is desired, write appropriate values to the modem control register (MCR).
6. Different interrupts can be generated by setting appropriate bits in Interrupt Enable register.
7. By setting DMA_MODE in FIFO control register, DMA transfer can be activated.

5.13 Serial peripheral interface (SPI)

The NNC_EPC001 has three SPI's (SPI0, SPI1 & SPI2), which act as either slave or master in full-duplex, half-duplex and simplex communication modes. The 3-bit baud rate input generator supports 8 master mode frequencies and the frame size is configurable from 1 bit to 16 bits. They have separate transmit/receive 16x16bits FIFOs which supports continuous data transmission in FIFO mode. The SPI clock polarity, clock phase and data shift order are user programmable.

The SPI interface support multi-slave communication using four slave chip-select (nss0~nss3). It also supports DMA transfer.

5.13.1 User Configuration

The configuration procedure is almost the same for master and slave. When a standard communication is to be initialized, perform these steps:

1. Write proper GPIO alternate registers: configure GPIO for MOSI, MISO, SCK and NSS pins.
2. Configure interrupts in interrupt enable register.
3. Write to the SPI_CTRL2 register:
    a. Configure the CHAR_LEN[3:0] to select the data length for the transfer.
    b. Select NSS port by configuring NSS0_EN, NSS1_EN, NSS2_EN or NSS3_EN.
    c. Select proper RX data sample phase of master by configuring SAMP_PHASE[1:0].
    d. Select proper C2T/T2C delay based on slave device requirement using C2T_DELAY/T2C_DELAY.
    e. Enable or disable TX/RX DMA in FIFO mode by configuring TXDMA_EN/RXDMA_EN.
4. Write to the FIFO Control register:
    a. Configure TX_FIFO_TH or RX_FIFO_TH to define the trigger level threshold.
    b. Clear TX/RX FIFO by setting TX_FIFO_CLR/RX_FIFO_CLR.
    c. Enable or disable FIFO mode using FIFO_EN bit.
5. Write to the SPI_CTRL1 register:
    a. Configure the serial clock baud rate using the BAUD_RATE[2:0].
    b. Configure clock polarity and phase using CPOL and CPHA.
    c. Select full-duplex or half-duplex or simplex communication mode by configuring BIDI_EN, BIDI_MODE and UNIDI_MODE appropriately.
    d. Configure the LSB_SEL to define the frame format or data shift order.
    e. Select NSS control mode by configuring NSS_TOGGLE, NSS_MST_CTRL, NSS_MST_SW.
    f. Select master or slave mode by configuring the MST_SLV_SEL.
    g. Enable SPI by setting SPI_EN.
6. Any transmit data should be loaded in Transmitter Holding register (THR) and any received data should be read from Receive Buffer register (RBR).

5.14 Inter-integrated sound interface (I^2S)

The NNC_EPC001 has two standard I2S interfaces (I2S0 and I2S1) which can operate as master or slave at half-duplex communication mode. They can be configured to transfer 16, 24 or 32 bits with 16-bit or 32-bit data resolution. They have a programmable clock polarity and they support most audio frequencies (8 kHz, 16 kHz, 22.05 kHz, 32 kHz, 44.1 kHz, 48 kHz, etc.). Data shift order is such that MSB is always send first.
By default, the I2S interface operates in Slave mode. Clock and Word-select signals are generated by master.

5.14.1 User Configuration

1. Write proper GPIO alternate registers: configure GPIO for SD, WS and SCK pins.

2. Configure interrupt in I2S interrupt enable register (I2S_IER)
3. Select the data and packet format in I2S Configuration Register (I2S_CFGR)
4. Set I2S_CFGR.MSTR to 1 for master mode; set I2S_CFGR.MSTR to 0 for slave mode
5. For transmission, follow further steps below:
    a. Set I2S_CFGR.TXRN to 1
    b. Write the transmit data to data register (I2S_DR)
    c. The first data written to Tx buffer is the left channel data, and followed by the right channel data
    d. To disable I2S, wait for TXE = 1 in the status register before writing I2S_CR.EN to 0
6. For reception, follow further steps below:
    a. Set I2S_CFGR.TXRN to 0
    b. Wait for RXNE = 1 in the status register, which indicates data has received in the RX buffer. Number of valid packets is shown in FILLED_ENTRIES in register I2S_SR.
    c. Read the received data from data register (I2S_DR)
    d. To disable I2S, write I2S_CR.EN to 0, then no more data will be written to RX_FIFO.
7. NOTE: By setting I2S_CFGR.DMA_EN, DMA transfer can be activated.

5.15 Quad-Serial peripheral interface (QuadSPI)

The QuadSPI is a specialized communication interface targeting single, dual or quad SPI Flash memories. It can operate in any of the three following modes:

- Indirect mode:
  All the operations are performed using the QuadSPI registers
- Status polling mode:
  The external Flash memory status register is periodically read and an interrupt can be generated in case of flag setting
- Memory-mapped mode:
  The external Flash memory is mapped to the device address space and is seen by the system as if it was an internal memory

In indirect mode, it supports a Flash memory capacity of up to 4GB. But in memory-mapped mode the addressable space is limited to 256MB. Both throughput and capacity can be increased two-fold using dual-flash mode, where two QuadSPI flash memories are accessed simultaneously.
The frame format and opcode are user programmable. There is an integrated FIFO (1 bytex32) for reception and transmission. DMA transfer is supported for indirect mode of operations.

5.15.1 User Configuration

Write proper GPIO alternate registers: configure GPIO for IO0, IO1, IO2, IO3, SCK and NCS pins

1. Indirect mode
    a. Set external SPI Flash memory size using FLASH_SIZE[4:0] in device configuration register (QUADSPI_DCR).
    b. Specify the number of data bytes to read or write in QUADSPI_DLR register.

c. Specify the frame format, mode and instruction code in the QUADSPI_CCR register. If FUNC_MODE[1:0] is set to 00, indirect write mode is selected and data can be sent to the Flash memory. If FUNC_MODE[1:0] is set to 01, indirect read mode is selected where data can be read from the Flash memory.
d. Specify optional alternate byte to be sent right after the address phase in the QUADSPI_ABR register.
e. Enable QUADSPI_EN together with different settings in control register (QUADSPI_CR)
f. Specify the targeted address in the QUADSPI_AR register.
g. Read/Write the data from/to the FIFO through the QUADSPI_DR register.

2. Status Polling mode
   a. Set external SPI Flash memory size using FLASH_SIZE in device configuration register (QUADSPI_DCR).
   b. This mode is enabled by setting the FUNC_MODE[1:0] field in QUADSPI_CCR to 10. In this mode, the programmed frame will be sent and the data retrieved periodically.
   c. The maximum amount of data read in each frame is 4 bytes. If more data is requested in QUADSPI_DLR, it will be ignored and only 4 bytes will be read.
   d. The periodicity can be specified in the QUADSPI_PIR register.
   e. Polling status Mask (QUADSPI_PSMKR) can be set in order to mask the status bytes received in polling mode.
   f. Enter the Polling status Match value (QUADSPI_PSMAR), which is used to compare with the received status.
   g. In case of match, the status match flag is set and an interrupt is generated if enabled, and the QuadSPI can be automatically stopped if the POLL_STOP bit is set.
   h. The latest retrieved status value can be read from QUADSPI_DR register.
3. Memory mapped mode
   a. Set external SPI Flash memory size using FLASH_SIZE in device configuration register (QUADSPI_DCR).
   b. This mode is entered by setting the FUNC_MODE to 11 in the QUADSPI_CCR register. Only read operations are allowed to the external Flash memory in this mode.
   c. Follow the frame settings of indirect read mode. Additionally, Prefetch mode and prefetch size can be configured in QUADSPI_CR register.
   d. The received data can be read from the memory mapped QuadSPI XiP device located at 0x7000_0000.

5.16 ECG module

The NNC_EPC001 includes an ECG Analog front-end, which is controlled by an AHB slave named as ECG controller. The ECG controller is for controlling the ECG Analog front-end, buffering the ADC data and processing the data by digital filters before sending to HeartApp. The data sending to Heartapp can be selected by user, which is either the data after digital CIC filter or the data after High Pass filter.

The ECG module supports ADC data rate from 62.5Hz to 32KHz. It supports three data formats, which are bipolar 2's complement, bipolar offset binary and unipolar. The measurement data from the output of each digital filter can be read by the MCU using AHB interface.

The ECG Analog part mainly includes the following blocks:

- PGA: It is a normal opamp with user-programmable gain
- Modulator: Second order sigma-delta modulator
- Reference: Reference voltage to ADC and system
- DC_LEAFOFF: DC Lead off detection function for monitoring the contact of the electrodes
- AC_LEADOFF: AC Lead off detection function for monitoring the contact of the electrodes
- RLD_DRIVER: Right leg driver signal
- LDO circuit: The Low Dropout Regulator supplies 3/3.3/3.6/3.9/4.2V for the whole ECG system

The ECG Controller digital part includes the following blocks:

- Sinc Filter: Digital CIC filter which converts 1-bit data from output of ADC modulator to 24-bits data
- FIFO: 32x8-bits FIFO for buffering ADC data
- ADC Control: To start ADC
- Clock Control: To generate ECG ADC clock, PGA/buffer chop clock and AC leadoff clock
- Low Pass filter for different cut-off frequency (9-11/15-20/25/50/100/150/200/350Hz)
- High Pass filter for different cut-off frequency (0.25/0.5/1/2.5/10/15/20/25Hz)
- Notch 50Hz band stop filter

Below figure illustrates the functional block diagram of ECG module.

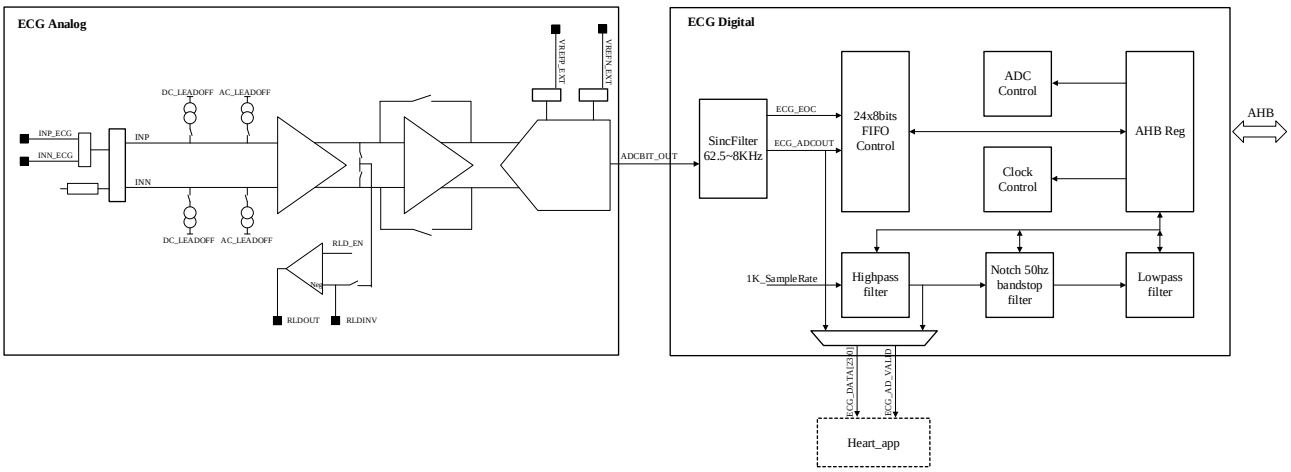


Figure 8 NNC-EPC001 ECG module

5.16.1 Specifications

The Table below shows the Specifications of ECG.

Minimum and maximum specifications apply from -40°C to +125°C.

Typical specifications are at +25°C, VOREG=3.5V, VDDD=1.5V

PARAMETR	CONDITION	MAIN VALUES			UNIT
		MIN	TYP	MAX	
Analog input					
Input Range		-vref/gain		+vref/gain	v
Input Cap			20		pf
DC input impedance			1000		MOHM
Input bias current			<1		nA
PGA					
PGA GAIN		1,2,4,6,8,12,60,120			
PGA BW	With 1nfcap	8			KHZ
ADC AND CHANNEL					
ADC resolution		24			BITS
ADC DATA RATE	256K FMod	62.5		8K	HZ
	1024K FMod	250		32K	HZ
Input noise	Gain=6,0.1~100HZ		2.5		uv(rms)
Gain error			0.1%		
Gain drift			10		ppm/℃
offset			100		uv
Offset error drift			8		uv/℃
CMRR	At 50hz		-110		db
DC LEADOFF					
DC current		25,50,100,200			nA
DC current accuracy			+/-5%		
Compare threshold		5%		95%	
AC LEADOFF					
Injection current		2,4,8,16			uA
Sink/source match			+/-5%		
Switch frequency		FMOD/2,FMOD/4			
Digital filter					
EMG_HPF	@ 1k data	10/15/20			HZ
EMG_LPF	@ 1k data	50/100/200/350			HZ
ECG_HPF	@ 1k data	0.05/0.25/0.5/1			HZ
ECG_LPF	@ 1k data	25/50/100/150			HZ
NOTCH(-60db)	@ 1k data	50			HZ

Table 11 NNC-EPC001 ECG Specifications

NOTE:
In the specification table, Fmod is the modulation frequency which is the ECG clock provided by digital interface to analog(D2A_ECG_CLK). This frequency is derived from system HF domain clock, which can be either 16MHz oscillator clock or PLL clock of maximum 64MHz. This ECG clock also depends on the divider value which can be set using ECG_CLK_DIV of System Control Clock Divider Register.
If HF clock is selected as 16MHz, for default clock divider settings the Fmod derived will be 256KHz. Whereas if PLL64MHz is selected as HF clock, then the Fmod derived will be 1024KHz.
The ADC data rate varies from Fmod/4096 to Fmod/32 based on the cic_rate setting in ECG ADC Control Register.

5.16.2 User Configuration

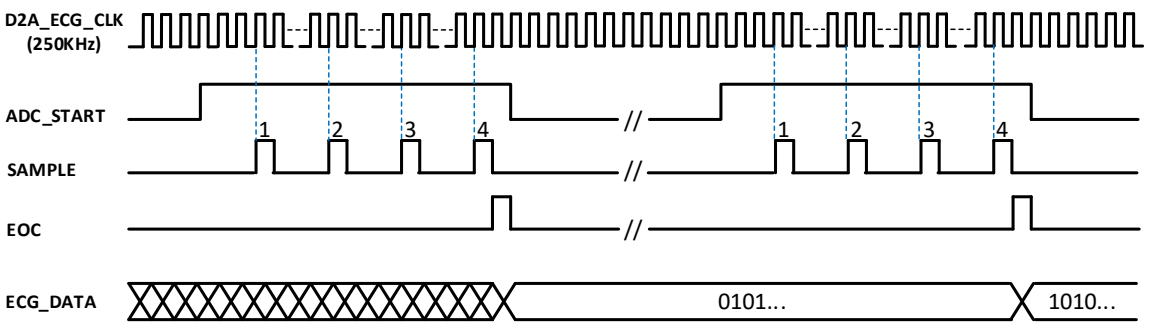
Perform initialization to enable Bandgap Buffer and supplies for ECG using ANA-BG Control and ANA-ECG_PPG Control registers, prior to configure ECG controller registers.

ECG Controller supports two Work Modes through WORK_MODE in ADC control Register:

- Normal Measurement mode
  The normal measurement mode will perform conversion in normal mode. The ECG normal measurement can be done either using single conversion or continuous conversion mode.
- Calibration mode
  The calibration mode will repeat the process of conversion and average the result based on the calibration number set by user in CALIB_NUM register. The calibration is done only using single conversion mode.

ECG Controller supports two conversion modes through CONV_MODE in ADC control Register:

1. Single Conversion mode:
   Setting the ADC_START bit initiates a single conversion, the ADC_START bit is automatically cleared after conversion completion. In single mode, 4 samples are required for each conversion and each time when conversion is completed, an EOC interrupt is generated. On interrupt, software can read conversion data from ECG_DATA register and then clear the EOC interrupt status.



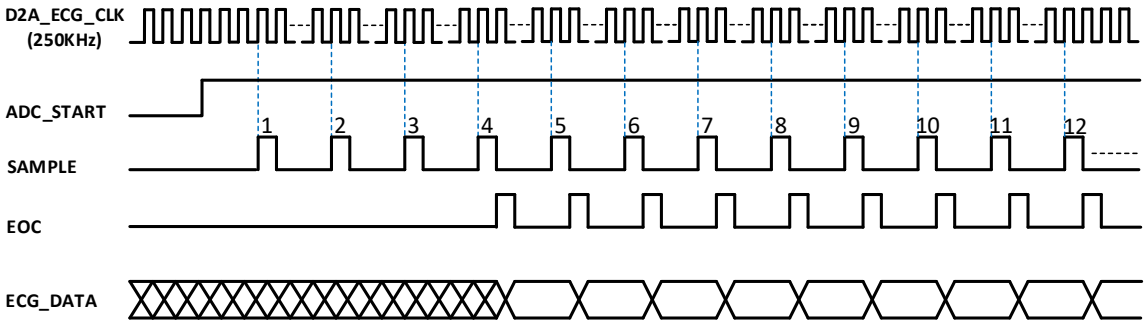
The single conversion mode does not require FIFO usage. Also, in this mode Low Pass/High Pass/Band Stop filters will not be used for processing.

Following are the steps to be followed in single conversion mode:

a. Configure interrupts in ECG Interrupt Enable register.
b. ECG Analog front-end settings should be done using ECG Configure 1 and 2 registers.
c. Set HEART_DATA_SEL = 0, so that data after CIC filter will be directed to HeartApp.
d. By default, the controller works in single conversion mode as CONV_MODE = 0 in ADC control register.
e. Select WORK_MODE=0 for measurement, WORK_MODE=1 for calibration.
f. Set the CIC_RATE, ADC_GAIN and Data format in ADC control register.
g. Enable the ADC and set ADC_START to 1, which starts conversion.
h. Wait for EOC_STS to read the measurement/calibration data from ECG Data register. This will read the data after digital CIC filter.

2. Continuous Conversion mode:

Conversion begins when ADC_START bit is set and continues until the ADC_START bit is cleared by software. This mode supports FIFO storage. If user enables FIFO mode, each time when conversion is completed, the data will be pushed to FIFO. When the FIFO filled entries are greater than or equal to threshold, FIFO threshold interrupt is generated so that software or DMA can read data from ECG_DATA register, and also when the FIFO filled entries are lesser than threshold, interrupt is cleared. Otherwise, if user doesn't enable FIFO mode, then each time when conversion is completed, an EOC interrupt is generated, and software can read data from ECG_DATA register and then clear the EOC interrupt status.



Note that in continuous conversion mode, the first 3 samples are ignored and actual conversion starts from 4th sample onwards. Also, in this mode Low Pass/High Pass/Band Stop filters are used for data processing.

Following are the steps to be followed in continuous conversion mode.

a. Configure interrupts in ECG Interrupt Enable register.
b. Enable FIFO and set threshold FIFO_TH in FIFO control register.
c. ECG Analog front-end settings should be done using ECG Configure 1 and 2 registers.
d. Use ECG Filter control register to set up digital High pass/ Low pass/ Band stop filters.

e. If HEART_DATA_SEL = 0, data after CIC filter will be directed to HeartApp. Otherwise, if HEART_DATA_SEL = 1, data after High Pass filter will be directed to HeartApp.
f. Set CONV_MODE = 1 in ADC control register, to select continuous conversion mode.
g. Set the CIC_RATE, ADC_GAIN and Data format in ADC control register.
h. Enable the ADC and set ADC_START to 1, which starts conversion.
i. Wait for FIFO threshold status to read ECG_DATA register till FIFO is empty.
j. Wait for HP_END_STS to read processed data after High Pass filter from ECG High Pass filter Data register.
k. Wait for BS_END_STS to read processed data after Band Stop filter from ECG Band Stop filter Data register.
l. Wait for LP_END_STS to read processed data after Low Pass filter from ECG Low Pass filter Data register.
m. In order to stop conversion, user need to clear ADC_START.

NOTE: Clearing the ADC_START before the conversion completion will immediately stops conversion and the digital CIC filter will be reset.

5.17 PPG module

The NNC_EPC001 includes a PPG Analog front-end, which is controlled by an AHB slave named as PPG controller. The PPG Controller provides user configurable signal pulses for the LED Driver of PPG Analog Front end system. It also does the buffering and processing of ADC data before sending to HeartApp. This block receives 8MHz clock and send it to analogue part for ADC sampling.

It has Four LED options to connect Red, Green, and IR LEDS with mode to turn on two LEDS out of four connected LEDs. i.e., PPG module supports 4-channel as well as 2-channel modes. The LED pulse-width is programmable to allow user for algorithm and power consumption optimization. It supports a Pulse-Repetition Frequency for LED driver from 62.5 Sps to 3861 Sps. It supports two data formats, which are bipolar 2's complement and bipolar offset binary. It has a dual data buffer connected to AHB to store PPG data after LPF. But there is option in PPG to select the data to HeartApp, which can be either data before or after LPF. It also has a power save feature to save power when ADC is not running.

The PPG module consists of the following major blocks:

- PPG Receiver:

It contains an Analog circuit to drive LEDs and to process the response signal from photo-detector. The digital circuit for PPG system provides user configurable signal pulses for this LED driver. It is the Pulse Generator in PPG digital circuit, which provides four pulse signals to drive each LED according to user configuration of pulses. The PPG receiver converts the current from the photodiode into voltage; and then digitizes using an incremental sigma-delta ADC which includes an incremental sigma-delta modulator and a 3rd digital COI filter. The digital COI filter converts 1-bit data from output of ADC modulator to 24-bits data. A digital low pass filter is included after obtaining ADC data.

Below figure illustrates the functional block diagram of PPG Receiver.

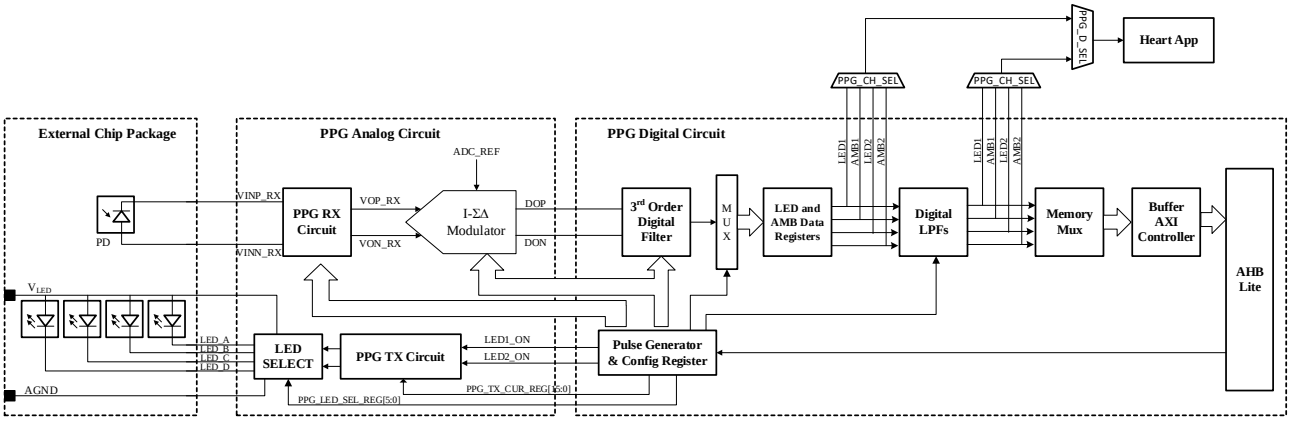


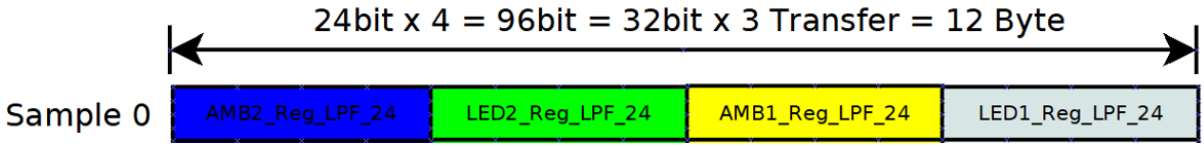
Figure 9 NNC-EPC001 PPG RX block

PPG module data are stored in dual-buffer as shown in block diagram, which are Memory 0 & Memory 1.
Initially Memory 0 buffer will start filling up with 48 bytes of PPG data samples, and then generates a trigger to MCU so that MCU can read data from first buffer. During MCU reading data from first buffer, the next 48 bytes of PPG data samples are written in second buffer, Memory 1, then again Memory 0 and so on. Each time when a buffer is full with 48 bytes, MCU receives a data ready interrupt, so that the data can be read out.
Each time MCU reads 32-bit data via AHB bus. The dual-buffer share the same AHB read address starting from 0x60300800.

NNC_EPC001 PPG operates in two modes.

- 4-channel mode

PPG data sample format in 4-channel mode is shown below.



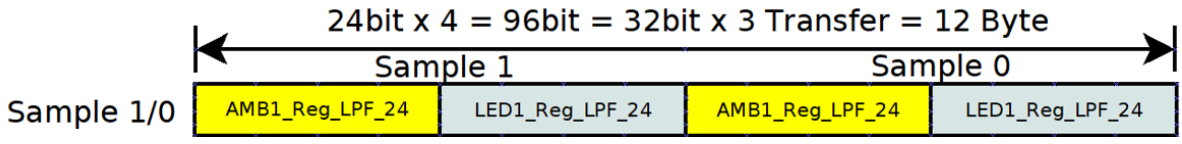
Data storage of PPG samples are illustrated below.

Memory0/Memory1 Location	Read address for MCU	PPG DATA 32 bits	4-Channel Mode
0	0x60300800	{AMB1[7:0],LED1[23:0]}	Sample 0
1	0x60300804	{LED2[15:0],AMB1[23:8]}	
2	0x60300808	{AMB2[23:0],LED2[23:16]}	
3	0x6030080C	{AMB1[7:0],LED1[23:0]}	Sample 1
4	0x60300810	{LED2[15:0],AMB1[23:8]}	
5	0x60300814	{AMB2[23:0],LED2[23:16]}	
6	0x60300818	{AMB1[7:0],LED1[23:0]}	Sample 2
7	0x6030081C	{LED2[15:0],AMB1[23:8]}	
8	0x60300820	{AMB2[23:0],LED2[23:16]}	
9	0x60300824	{AMB1[7:0],LED1[23:0]}	Sample 3
10	0x60300828	{LED2[15:0],AMB1[23:8]}	
11	0x6030082C	{AMB2[23:0],LED2[23:16]}	

Table 12 NNC-EPC001 PPG 4-channel data storage

- 2-channel mode

PPG data sample format in 2-channel mode is shown below.



Data storage of PPG samples are illustrated below.

Memory0/Memory1 Location	Read address for MCU	PPG DATA 32 bits	2-Channel Mode
0	0x60300800	{AMB1[7:0],LED1[23:0]}	Sample 0/1
1	0x60300804	{LED1[15:0],AMB1[23:8]}	
2	0x60300808	{AMB1[23:0],LED1[23:16]}	
3	0x6030080C	{AMB1[7:0],LED1[23:0]}	Sample 2/3
4	0x60300810	{LED1[15:0],AMB1[23:8]}	
5	0x60300814	{AMB1[23:0],LED1[23:16]}	
6	0x60300818	{AMB1[7:0],LED1[23:0]}	Sample 4/5
7	0x6030081C	{LED1[15:0],AMB1[23:8]}	
8	0x60300820	{AMB1[23:0],LED1[23:16]}	
9	0x60300824	{AMB1[7:0],LED1[23:0]}	Sample 6/7
10	0x60300828	{LED1[15:0],AMB1[23:8]}	
11	0x6030082C	{AMB1[23:0],LED1[23:16]}	

Table 13 NNC-EPC001 PPG 2-channel data storage

5.17.1 Specifications (PPG Rx)

The Table below shows the Specifications of PPG Receiver.

Parameters	Min.	Typ.	Max	Unit
Number of Photodetector/ input channel (System on Package)		1		
Total integrated Input Referred Noise current	5.3		900	pArms
Input Capacitance			1000	pF
Input PD Current Range, I_{PD}	0.5		50	µA
Dynamic Range (dB)		>102		dB
Supply Voltage (V)	3.0	3.3	3.6	V
Current Consumption		<430		µA
Standby Current		< 3		µA

Table 14 NNC-EPC001 PPG RX Specifications

- PPG Transmitter:

It consists of a current DAC with programmable LED current, H-bridge/Push-pull driver, and MUX for selecting the input channel/LEDs. The PPG transmitter allows four LED options to connect Red, Green, and IR LEDs along with a mode to turn on two LEDs out of four connected LEDs. It provides the programmable LED current from 1mA to 100mA for each LED with 8-bit DAC resolution.

Below figure illustrates the functional block diagram of PPG Transmitter.

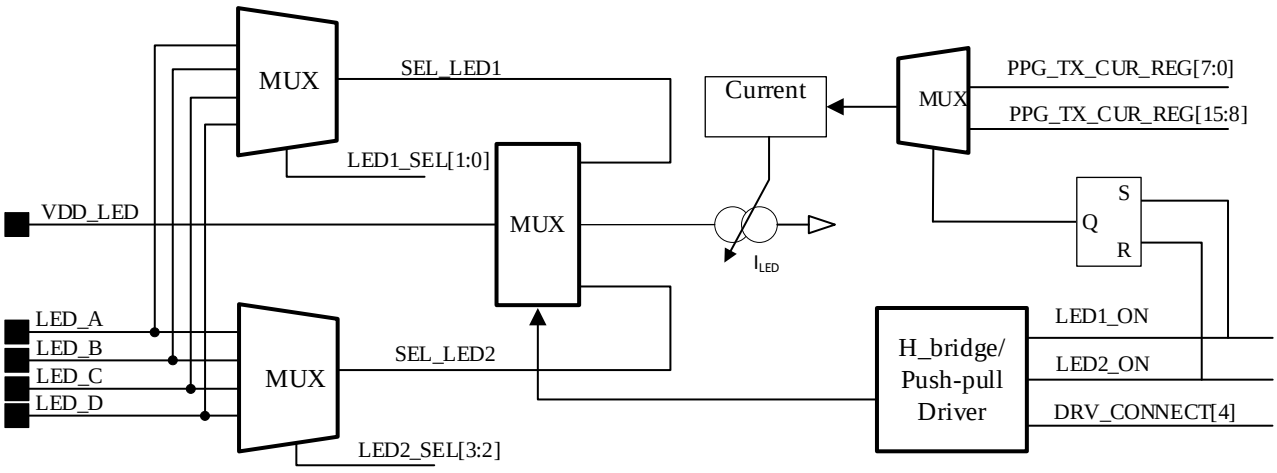
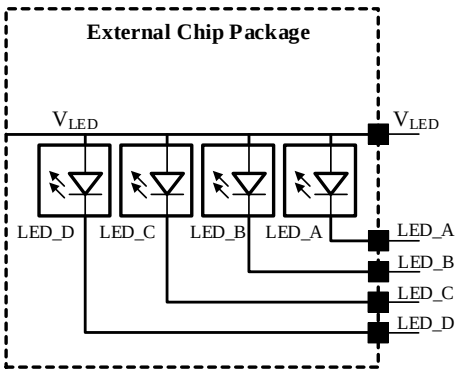


Figure 10 NNC-EPC001 PPG TX block

The LED connection of PPG transmitter can be of two types, either Push-pull or H-Bridge connection.

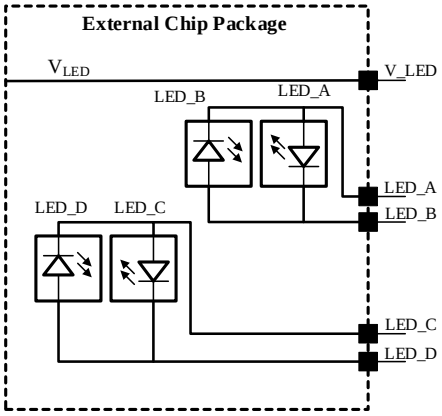
- Push-pull Connection



PPG_LED_SEL [5:0]	LED1 connection	LED2 Connection
10 00 00	Not Valid	Not Valid
10 00 01	LED A	LED B
10 00 10	LED A	LED C
10 00 11	LED A	LED D
10 01 00	LED B	LED A
10 01 01	Not Valid	Not Valid
10 01 10	LED B	LED C
10 01 11	LED B	LED D
10 10 00	LED C	LED A
10 10 01	LED C	LED B
10 10 10	Not Valid	Not Valid
10 10 11	LED C	LED D
10 11 00	LED D	LED A
10 11 01	LED D	LED B
10 11 10	LED D	LED C
10 11 11	Not Valid	Not Valid

For Push-pull connection shown above, the PPG_LED_SEL_REG [4:0] values are valid.

- H-Bridge Connection



PPG_LED_SEL [5:0]	LED1 connection	LED2 Connection
11 00 01	LED A	LED B
11 01 00	LED B	LED A
11 10 11	LED C	LED D
11 11 10	LED D	LED C

For H-bridge connection shown above, the PPG_LED_SEL_REG [4:0] values are valid. Other valid values are possible for the appropriate H-bridge connections.

5.17.2 Specifications (PPG Tx)

The Table below shows the Specifications of PPG Transmitter.

Parameters	Min.	Typ.	Max	Unit
Number of LEDs, NLEDs (System on package)		4		
Allowable Number of LEDs for operation		2		
Mode to fire two LEDs in parallel		Yes		
Drivers support H-bridge/Push-Pull		Both H-bridge and Push-Pull		
Output Current Range	0		100	mA
Output Current Resolution		8		Bits
Output Current Accuracy		±10		%
Pulse Repetition Frequency, PRF	62.5		3861	SPS
Pulse Duty Cycle, PDC	1		25	%
Pulse width	62.5		2000	µs
Rise Time or Fall Time			10	µs
Supply Voltage	3.0	4.0	5.25	V
Supply Current		LED Current		µA
CMMR		≥110		dB
PSRR		≥80 (@50 Hz, 60 Hz) 60 (@0.1Hz to 20Hz)		dB

Table 15 NNC-EPC001 PPG TX Specifications

- PPG Voltage reference generator:

The reference signal generated by this block is used for common mode voltage (Vcm) for PPG Receiver, reference (Vrefp, Vrefn) and common-mode voltage (Vcm) for sigma-delta ADC. This circuit requires trimming bits (VREF_TRIM<5:0>), provided from digital interface in order to adjust the DC voltage level of Vcm, Vrefp, and Vrefn.

- LDO circuit:

It supplies 3.3 v to the whole PPG receiver system.

5.17.3 PPG system configuration

The Numbers are calculated for Clock Frequency (Fs) of 8.0 MHz.

Variable	Description	Minimum	Maximum	Unit
FCP	Total number of clocks in a full conversion period	2072	128000	CLK
N_{L1}/ N_{A1}	LED1/AMB1 on time (1)	518	518 x m (2)	CLK
N_{L2}/ N_{A2}	LED2/AMB2 on time (1)	518	518 x n (2)	CLK
$NL1_{RxTime}$	LED1 sample time	402	N_{L1} – 116	CLK
$NA1_{RxTime}$	AMB1 sample time	402	N_{A1} – 116	CLK
$NL2_{RxTime}$	LED2 sample time	402	N_{L2} – 116	CLK
$NA2_{RxTime}$	AMB2 sample time	402	N_{A2} – 116	CLK
$NL1_{AVG}$	Number of averages for A2D conversion of LED1	1	$(N_{A1}\div 518)-1$ (3)	
$NA1_{AVG}$	Number of averages for A2D conversion of AMB1	1	$(N_{L2}\div 518)-1$ (3)	
$NL2_{AVG}$	Number of averages for A2D conversion of LED2	1	$(N_{A2}\div 518)-1$ (3)	
$NA2_{AVG}$	Number of averages for A2D conversion of AMB2	1	$(N_{L1}\div 518)-1$ (3)	

Table 16 NNC-EPC001 PPG system settings

****Restriction of Input parameters**:

(1) $N_{L1}=N_{A1}$ and $N_{L2}=N_{A2}$. These numbers are set to be multiples of 518, which is due to the filter characteristics.

(2) The selection of the interger numbers m and n must meet 2 conditions:

1. 1<= m,n <= 122
2. $N_{L1} + N_{A1} + N_{L2} + N_{A2}$ <= FCP

(3) The Analog to Digital (A2D) conversion of LED1 occurs during the AMB1 on time, the Analog to Digital (A2D) conversion of AMB1 occurs during the LED2 on time, and so on.

The Pulse Repetition Frequency (PRF) is from from 62.5 Sps to 3861 Sps, which is calculated by PRF = Fs/FCP. The selection of FCP should take into account the PRF value that the user desires, based on that formula.

5.17.4 User Configuration

Perform initialization to enable Bandgap Buffer and supplies for PPG using ANA-BG Control and ANA-ECG_PPG Control registers, prior to configure PPG controller registers

Follow the steps below to perform PPG measurement.

1. Configure No: of clocks in a PRF period = (Fs/PRF) in the PPG_PRF_REG register
2. Set NL1 period in PPG_LED1_ON_REG register
3. Set NL2 period in PPG_LED2_ON_REG register
4. Set LED1/AMB1 sample time in PPG_LED1_SAMP_REG register
5. Set LED2/AMB2 sample time in PPG_LED2_SAMP_REG register
6. Set LED1/AMB1/LED2/AMB2 average value in PPG_LED_AVG_REG register
7. Configure LED1 and LED2 current selection in PPG_TX_CUR_REG register
8. Select 4-channel mode or 2-channel mode using LED_NUM_CH of PPG_LED_SEL_REG register
9. Configure the LED selection for Push-pull/H-bridge connection appropriately in PPG_LED_SEL_REG register
10. Use PPG_D_FORMAT of PPG_DATA_CONF_REG to select either 2's complement or offset binary data format
11. If PPG_D_SEL = 0, data before Low pass filter will be directed to HeartApp. Otherwise, if PPG_D_SEL = 1, data after Low pass filter will be directed to HeartApp
12. Configure PPG Receiver Analog front-end (AFE) Gain settings in PPG_RX_CONF_REG register. This register has a PWRSAVEN bit, which can be enabled to save ADC power. Also enable the PPG receiver AFE circuits by setting PPGRXEN bit of this register
13. Enable PPG RX digital control block using PPG_RX_EN_REG register
14. Make sure interrupt is enabled, so that MCU can wait for data ready interrupt before reading data from PPG data buffer.

5.17.5 Timing Diagram

Pulse Generator generates control signals for PPG Analog-front end as per the following timing diagram. These signals are generated based on the user configuration of different registers as per *Section* 5.17.3.

LED1_ON (N_{L1}): It is generated based on Pulse-Repetition Frequency (PRF) and Duty Cycle (D1) of LED1/AMB1

LED2_ON (N_{L2}): It is generated based on Pulse-Repetition Frequency (PRF) and Duty Cycle (D2) of LED2/AMB2

SAMP: SAMP_ signals are generated based on LED sample settings

CONV: CONV_ signals are generated based on LED average settings

ADC_CLK: ADC_CLK is an 8MHz clock signal which is reset on ADC_RST

ADC_RST: ADC_RST occurs at one ADC_CLK delay after falling edge of SAMP signal

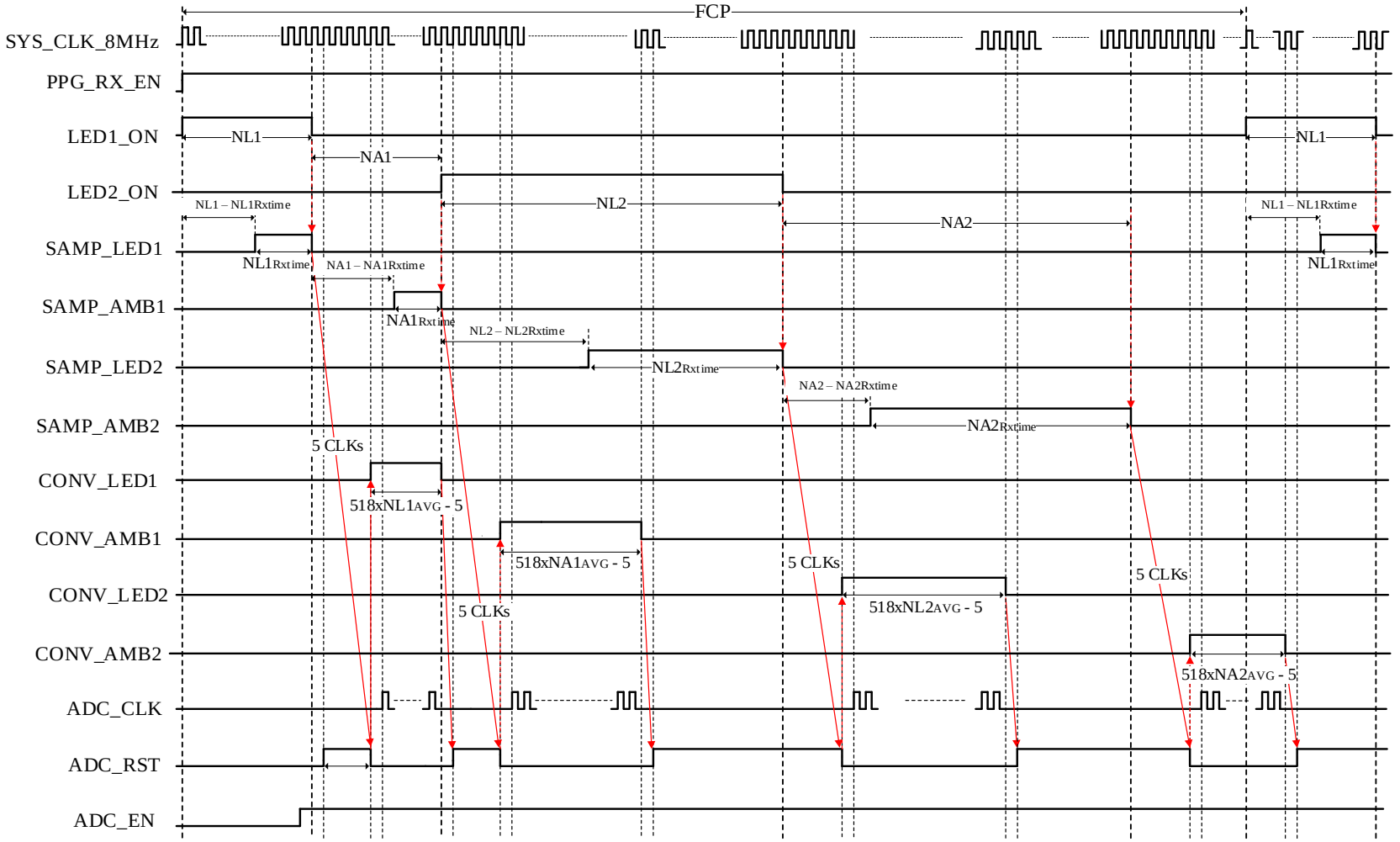


Figure 11 NNC-EPC001 PPG Timing Diagram

5.18 HeartApp module

The HeartApp module receives ADC samples from ECG module, which will be put on for further processing using FIR filters in order to extract/measure the following parameters.

- ECG (Electro-cardiogram) can measure:
  - HR (Heart Rate)
  - RR (Respiratory Rate)
  - SNA (Anxiety Index)
  - ARR (Arrhythmia)
  - HRV (Heart Rate Variability)
  - QT (interval)

5.18.1 User Configuration

1. ECG data moving average can be enabled or disabled using Heart Application Control register (HA_CR)
2. Number of data samples that should be skipped before measurement can be configured in MEASURE_PRE of HA_CFGR register
3. The interrupt which indicates end of measurement can be enabled in Heart Application Interrupt Enable register (HA_IER)
4. When interrupt occurs, following parameters can be read from Heart Application Data register 0 (HA_DR0):
   a. Wave too small flag
   b. Arrhythmia index
   c. Respiratory Rate
5. When interrupt occurs, following parameters can be read from Heart Application Data register 1 (HA_DR1):
   a. Anxiety Index
   b. ECG Heart Rate result
6. When interrupt occurs, following parameter can be read from Heart Application Data register 2 (HA_DR2):
   a. QT result

5.19 JTAG debug system

A single external industry-standard 1149.1 JTAG interface is provided to test and debug the system. It is a standard 4 wire JTAG interface with TCK, TMS, TDI, TDO. This enables access to the internal S2 debug module.

6. Memory mapping

The table below shows S2 Global address map.

Start	End	Attribute	Description
0x0000_0000	0x0000_0FFF		Debug
0x0000_1000	0x0000_2FFF	Reserved	
0x0000_3000	0x0000_3FFF	RWX A	Error Device
0x0000_4000	0x01FF_FFFF	Reserved	
0x0200_0000	0x02FF_FFFF	RW	CLIC
0x0300_0000	0x1FFF_FFFF	Reserved	
0x2000_0000	0x3FFF_FFFF	RWXI	Peripheral Port (512 MiB)
0x4000_0000	0x5FFF_FFFF	RWXI	System Port 0 (512 MiB)
0x6000_0000	0x7FFF_FFFF	RWXI	System Port 1 (512 MiB)
0x8000_0000	0x8000_7FFF	RWX	TIM 0 (32 KiB)
0x8000_8000	0x8000_FFFF	RWX	TIM 1 (32 KiB)
0x8001_0000	0xFFFF_FFFF	Reserved	
**R - Read, W - Write, X - Execute, I – Instruction Cacheable, A – Atomics			

Table 17 NNC-EPC001 Global address map

The following Figure below shows the detailed memory and peripheral address space connected to AHB lite ports.

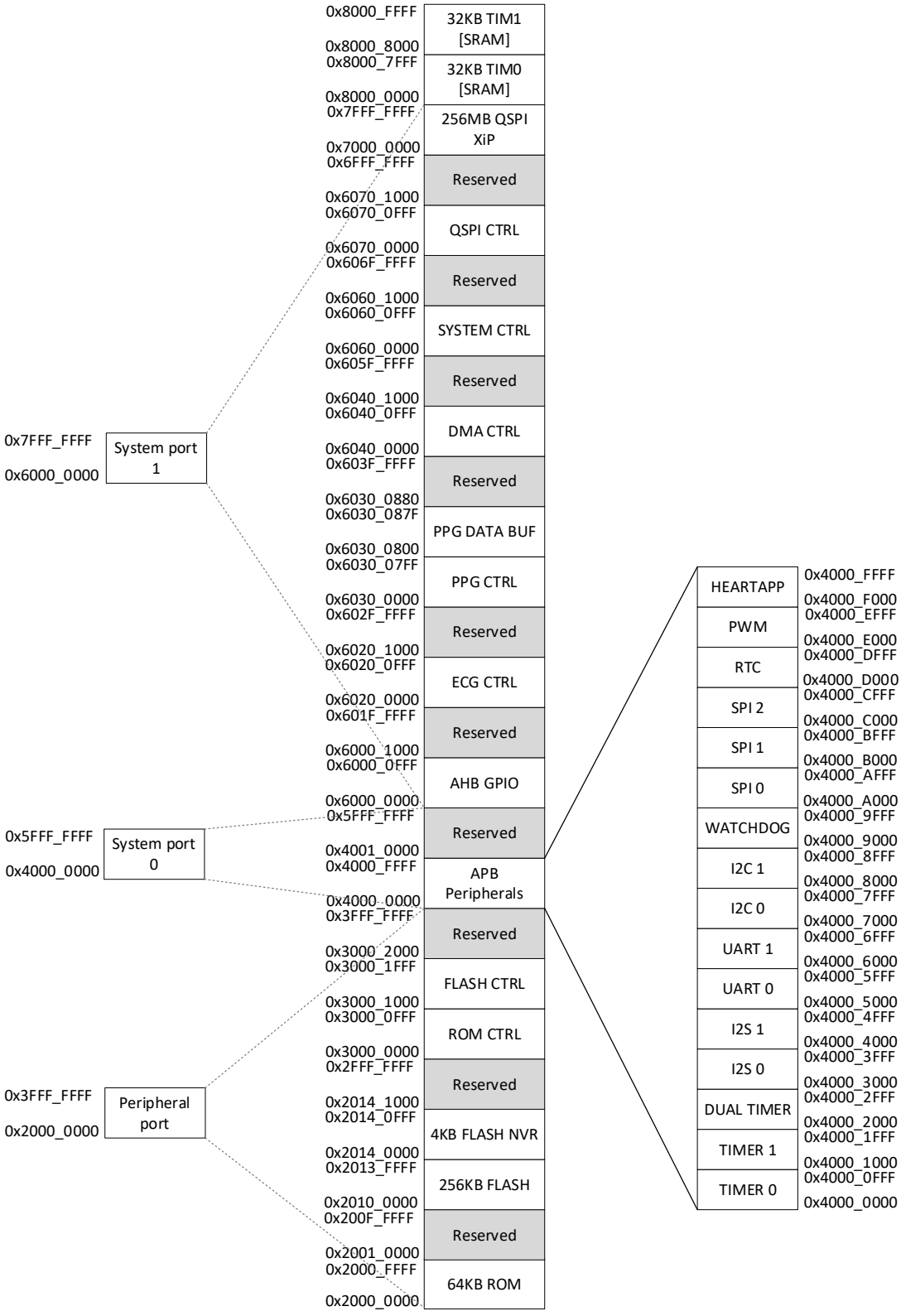


Figure 12 NNC-EPC001 Memory Map

7. Application Example

The Application schematic diagrams in *Section* 7.2 shows an application example using NNC-EPC001 device to measure ECG and PPG signals. The features implemented are:

- DVDD (1.5V) power supply initially generated by LDO15 linear regulator. User can switch the power supply source for digital part of the chip using the DCDC15 buck regulator when the system is in normal status.
- ECG circuit with a few components added into the INN_ECG and INP_ECG made it available where application PCB space is limited. User can design the on-board pads to sense ECG signals or they can be connected through a header.
- The proposed PPG circuit accepts two types of LED TX signals including bridge mode and push pull mode.
- There are multiple GPIO inputs/outputs, alternate functions with high-speed SPI, I2C, I2S, and UARTs that provide flexible options for any specific application.

7.20 List of Components

Table below shows the component list used in the application circuit.

Components	Value	Quantity
C1, C2	22pF	2
C3, C4	15pF	2
C5, C8, C12, C13, C22, C24, C25, C27, C28, C30, C31, C32, C38, C39, C40, C41, C42, C43, C44, C45, C46, C48, C50, C51, C54, C56	100nF	26
C6, C10, C19, C20, C21, C23, C26, C55	1uF	8
C7, C9, C14, C15, C17, C18	10nF	6
C11, C16	1nF	2
C29, C33, C34, C35, C36, C37, C47, C49, C52, C53	10uF	10
FB1	120R	1
J4	ZX62-B-5PA(11)	1
L2	22uH(NC)	1
L3	22uH SMD 1365 5.7A 37mOhm	1
LED1	BL-HG836D-TRB	1
P1	PJ-328-5P	1

R1, R2, R3, R4	33R	4
R5	5.1M	1
R6, R7, R11, R12	0.0R(NC)	4
R8, R9	20K	2
R10	0.0R	1
R13	51K	1
R14	110K	1
R15, R33	1K	2
R16	137K	1
R17	62K	1
R18, R21, R22, R26, R27	NC	5
R19, R20	10M	2
R23, R25, R29	180K	3
R24	RA-PCB	1
R28	360K	1
R30	47k	1
R31	30k	1
R32	0.5R	1
R34, R35, R36	0.0R	3
RESET1	UK-B0228	1
S1, S2, S3, S4, S5, S6, S7	SS-12D03	7
SW1, SW2	DSIC02LSGET	2
U1	EPC001_88	1

Table 18 NNC-EPC001 Component list

7.21 Application Schematic

Power related schematic is shown as below.

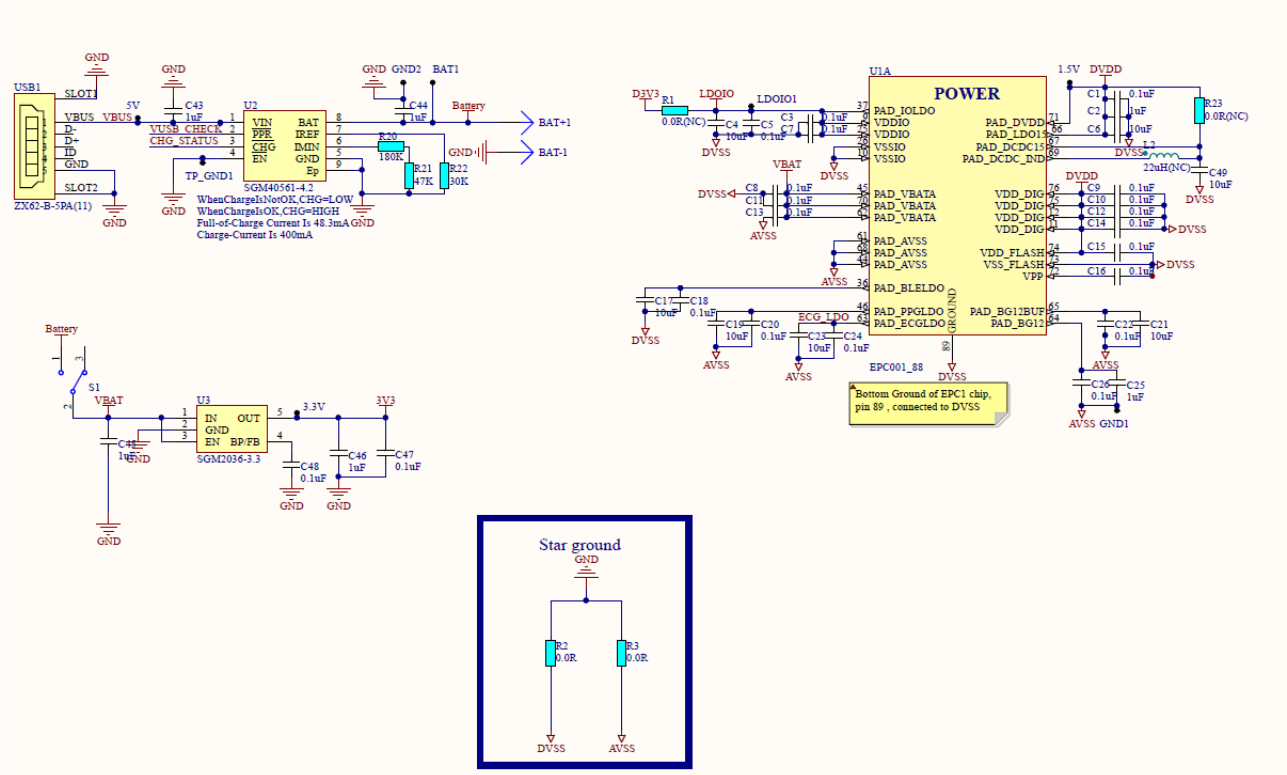


Figure 13 NNC-EPC001 Power Schematic

Clock and System schematic is shown as below.

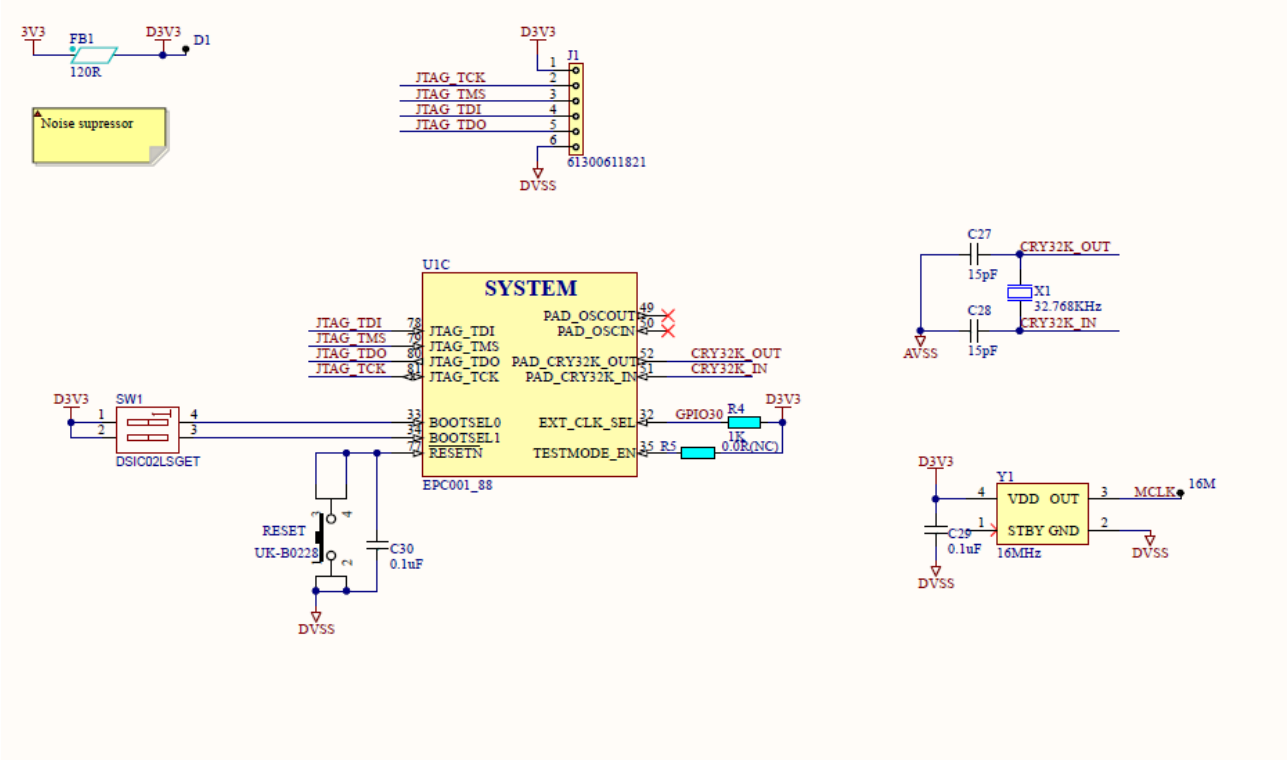


Figure 14 NNC-EPC001 System Schematic

IO schematic is shown as below.

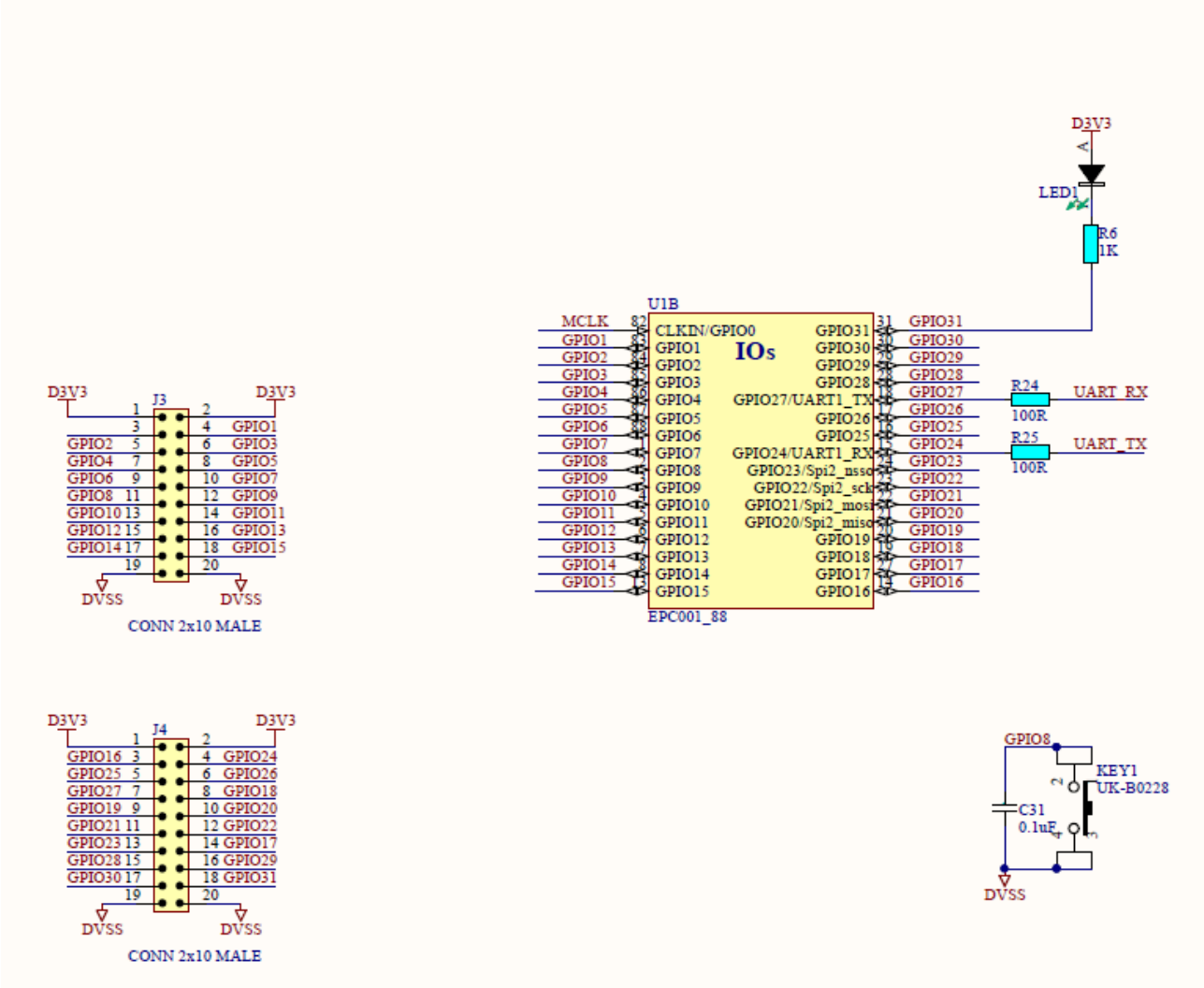


Figure 15 NNC-EPC001 IO Schematic

PPG schematic is shown as below.

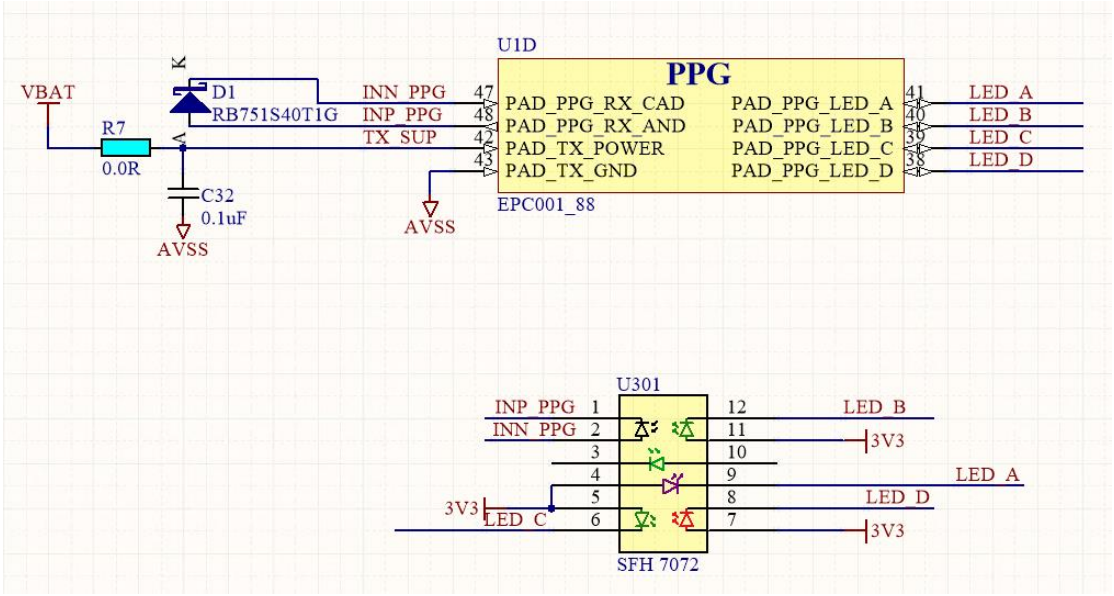
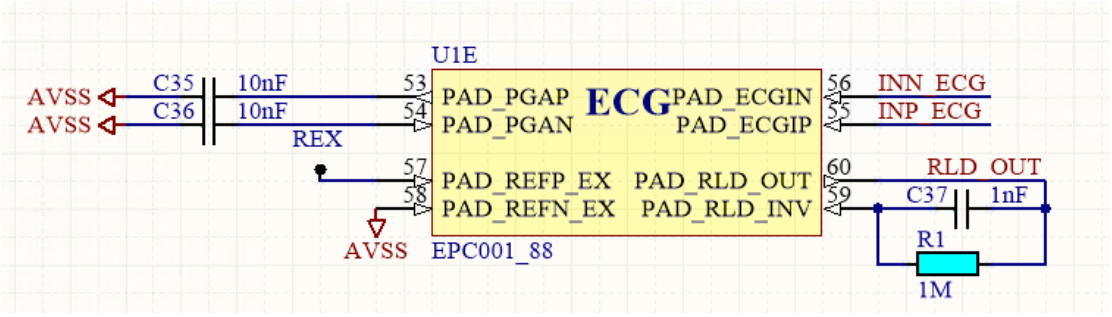
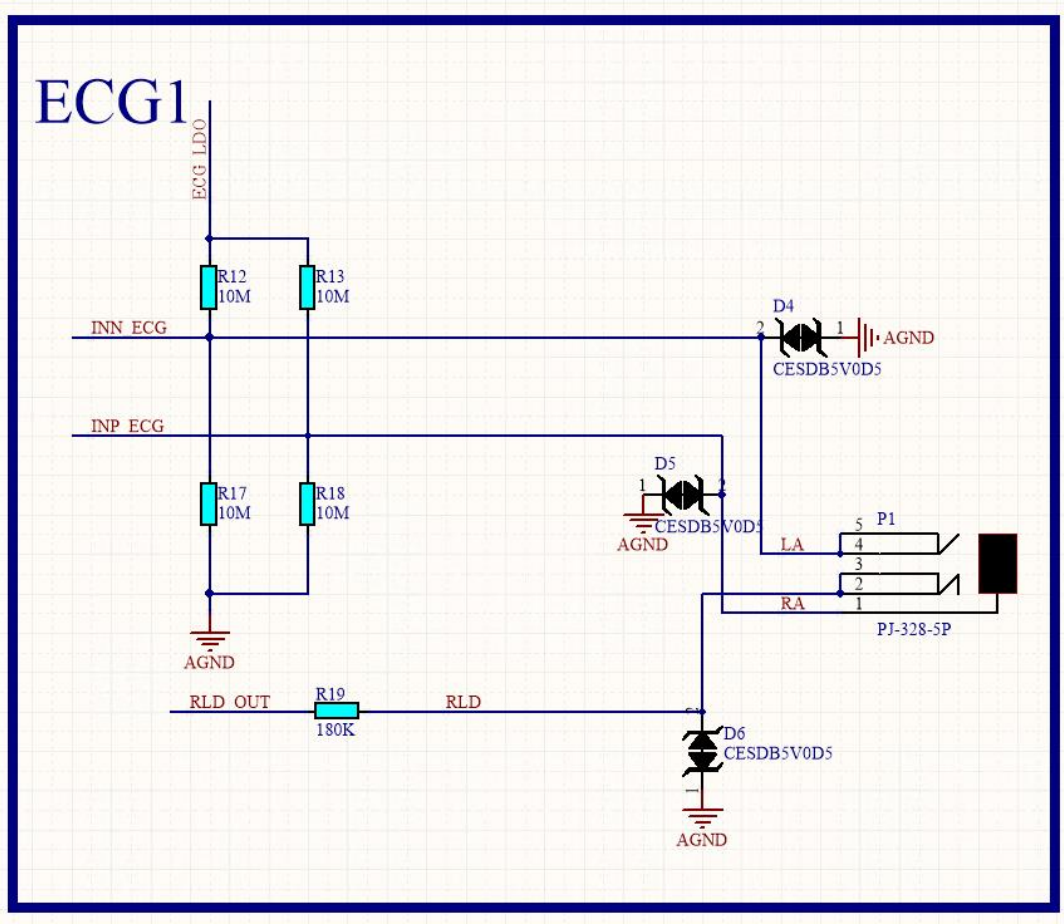


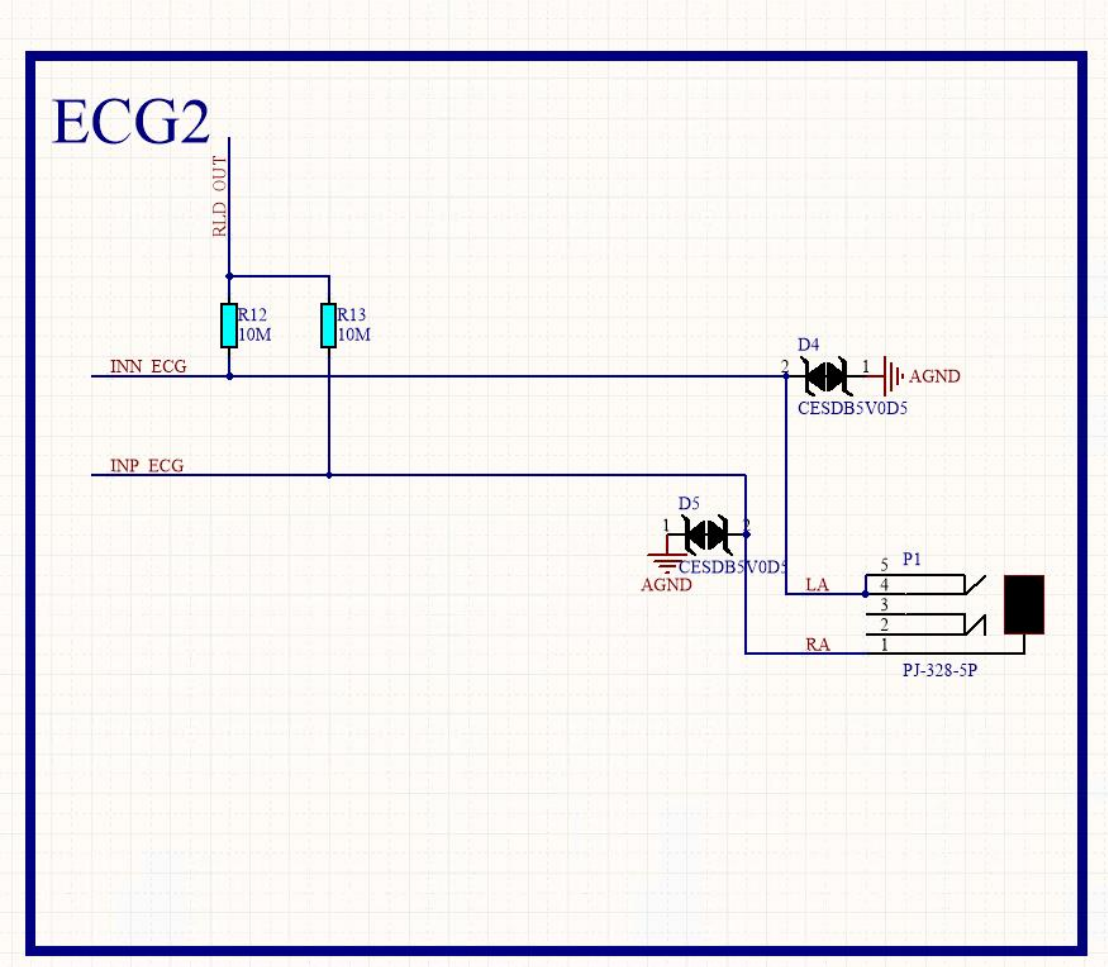
Figure 16 NNC-EPC001 PPG Schematic

ECG schematic is shown as below.



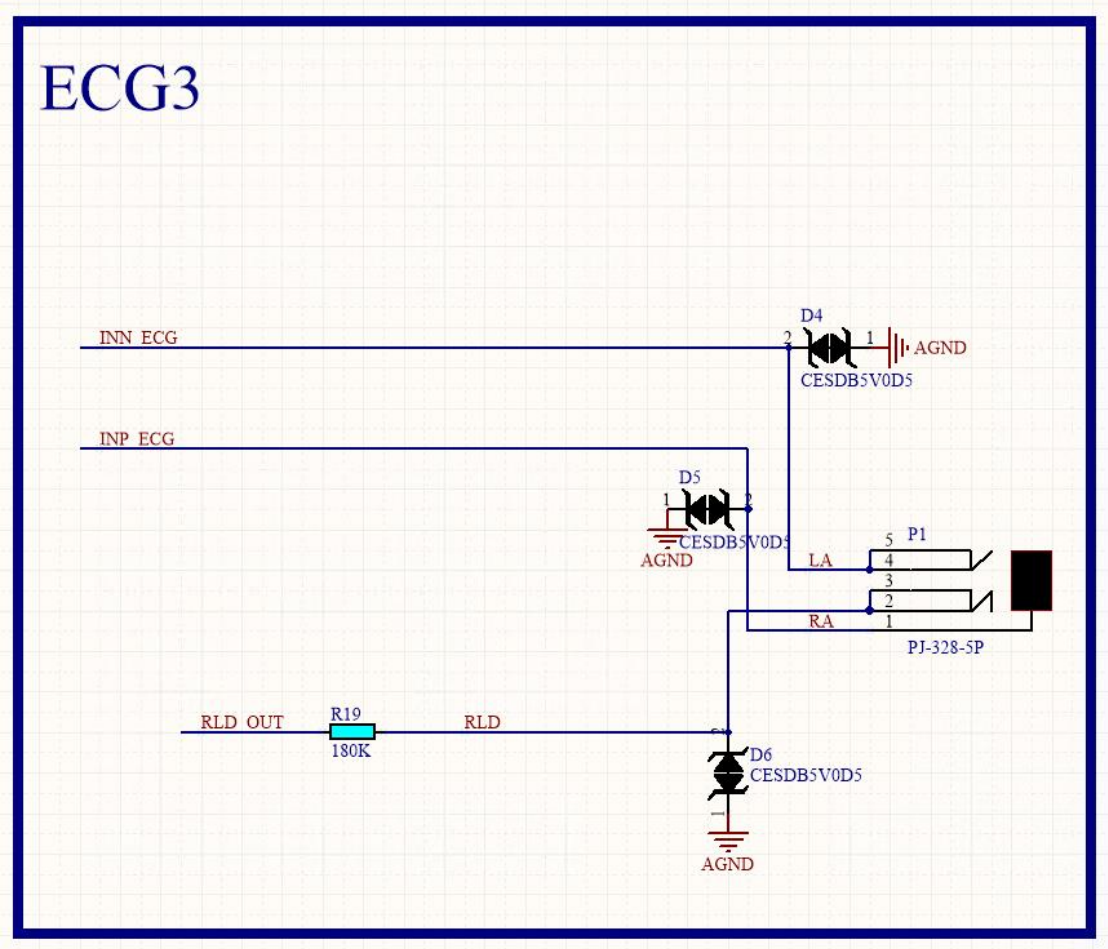


The ECG1 schematic shows four 10 MΩ resistors to set the mid-scale reference voltage. The advantage of this connection is that three electrodes and two electrodes can be compatible at the same time, but the disadvantage is that the input impedance is reduced. For applications requiring input impedance, please refer to ECG2 and ECG3 schematics.



For wearable exercise devices, EPC001 is typically placed in a pod near the heart. The two sensing electrodes are placed under the pectoral muscles; no driven electrode is used. Because the distance from the heart to EPC001 is short, the heart signal is strong and there is less muscle artifact interference.

In this configuration, space is at a premium. By using as few external components as possible, the circuit in ECG2 schematic is optimized for size. A shorter distance from EPC001 to the heart makes this application less vulnerable to common-mode interference. However, since RLD driver is not used drive an electrode, it can be used to improve the common-mode rejection by maintaining the mid-scale voltage through the 10 MΩ bias resistors.



EPC001 uses a three-electrode patient interface. The RLD circuit drives to the third electrode, which can also be located at the hands, to cancel common-mode interference.

Figure 17 NNC-EPC001 ECG Schematic: EPC001's connections and ECG1, ECG2, ECG3 schematics

8. Package information

8.1 QFN88L

NNC_EPC001 QFN88L package PCB footprint is shown below.

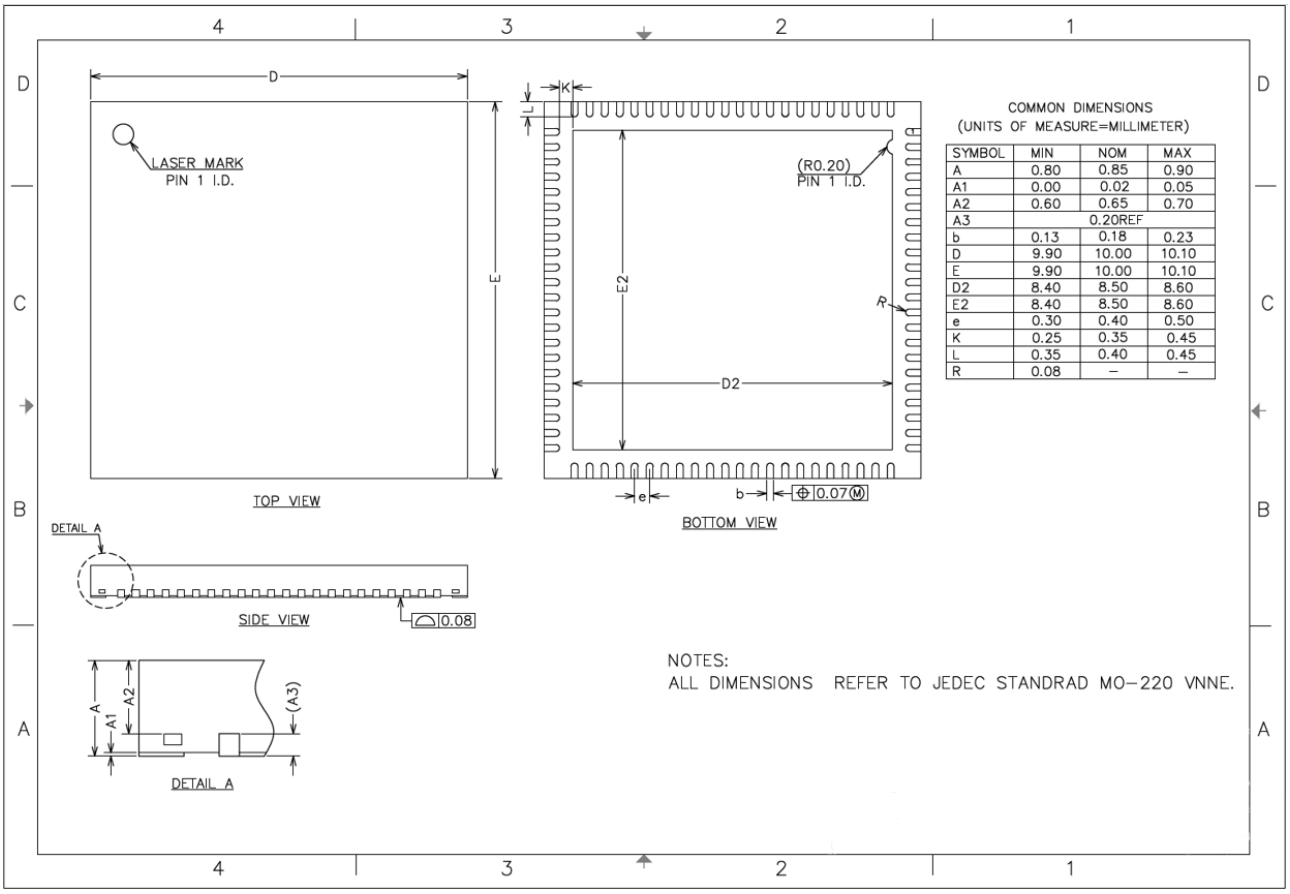
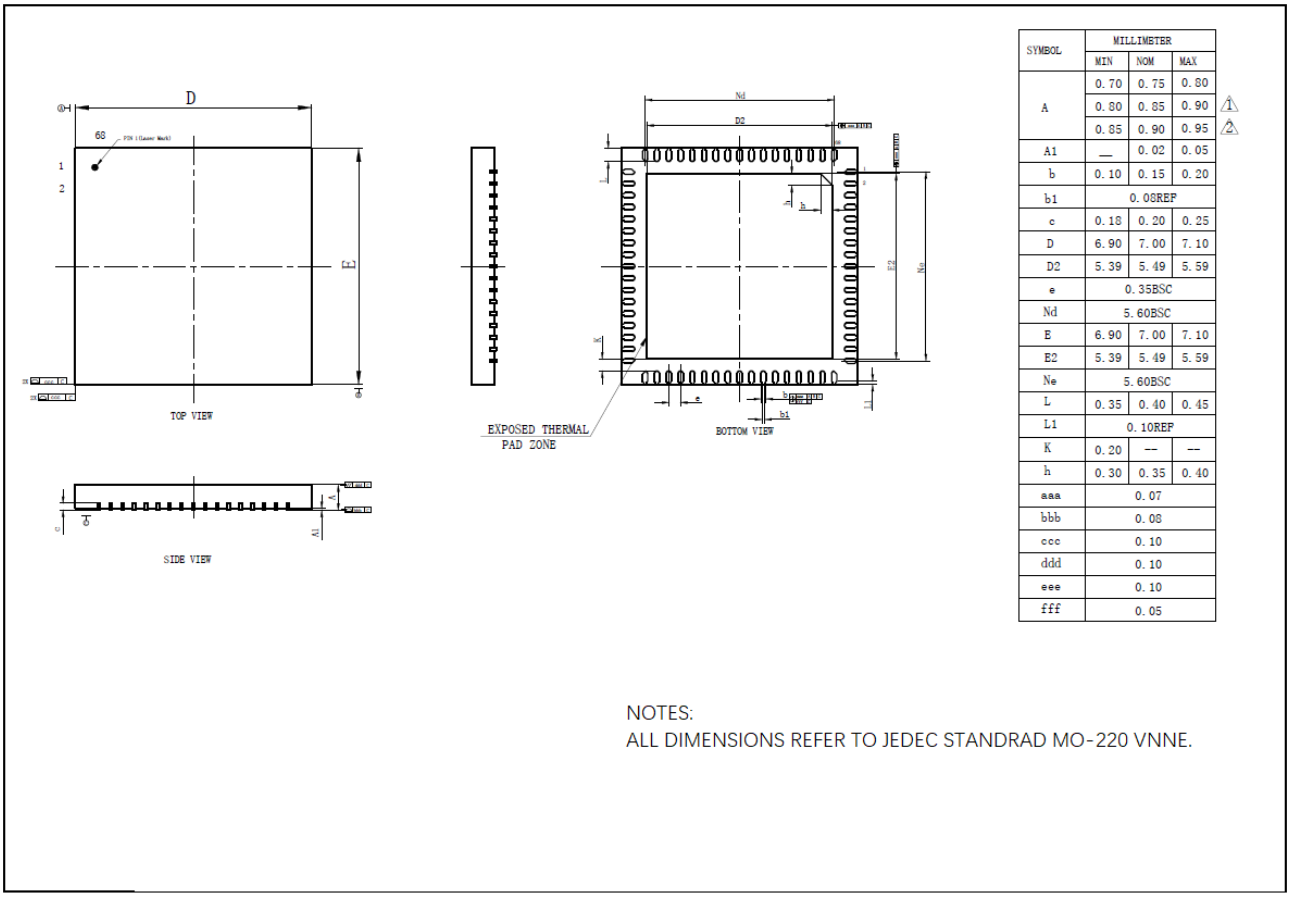


Figure 18 NNC-EPC001 QFN88L package outline

8.2 QFN68L

NNC_EPC001 QFN68L package PCB footprint is shown below.



SYMBOL	MILLIMETER		
	MIN	NOM	MAX
A	0.70	0.75	0.80
	0.80	0.85	0.90
	0.85	0.90	0.95
A1	—	0.02	0.05
b	0.10	0.15	0.20
b1	0.08REF		
c	0.18	0.20	0.25
D	6.90	7.00	7.10
D2	5.39	5.49	5.59
e	0.35BSC		
Nd	5.60BSC		
E	6.90	7.00	7.10
E2	5.39	5.49	5.59
Ne	5.60BSC		
L	0.35	0.40	0.45
L1	0.10REF		
K	0.20	—	—
h	0.30	0.35	0.40
aaa	0.07		
bbb	0.08		
ccc	0.10		
ddd	0.10		
eee	0.10		
fff	0.05		

NOTES:
ALL DIMENSIONS REFER TO JEDEC STANDRAD MO-220 VNNE.

Figure 19 NNC-EPC001 QFN68L package outline

9. Order information

10. Acronyms

CMMR	Common Mode Rejection Ratio
PSRR	Power Supply Rejection Ratio

11. Revision history

Table 19 Revision history

Version	Date	Description	Author
Rev. A	2022 May 4th	Initial format	N.T.
Rev. B	2022 July 1st	First Draft	Ophina.C
Rev. C	2022 Sep 16 th	Added package QFN68L and update figures of Application Schematic	Michael
Rev. D	2022 Oct 17th	Modified PPG specs table and added acronym table	N.T.
Rev. E	2023 Dec 18th	Updated PPG and ECG application schematics	N.T.