Description

CambridgeIC's CAM622 is a Resonant Inductive Encoder IC. It includes a Sensing Engine for measuring contactless position, and an Interface Processor that can generate encoder style outputs.

The Sensing Engine connects to an external sensor. This detects the position of a contactless inductive resonant target. Both sensor and target are made from PCBs. They include coil patterns for resonant inductive sensing. These are available from CambridgeIC, so customers can manufacture the parts themselves and integrate them with their product. The CAM622 works with a variety of different sensor and target sizes.

The Interface Processor can generate encoder style digital outputs, using an SPI interface for configuration and diagnostics. A host may alternatively use SPI as the primary interface, reading measurement results at high speed.

Features

- Works with CambridgeIC's "Type B" sensor PCBs
- Fully ratiometric Sensing Engine
- Automatic tuning to target frequency
- Optional adaptive filter for extra resolution
- Filter can be configured for zero phase delay
- Encoder style ABN outputs with True Edge Timing
- ABN outputs absolute at start-up (NSTART=1)
- SPI for configuration or reading results
- BiSS interface support
- Internal software upgradable over SPI
- Exceptionally stable across temperature

Performance

- Up to 33000 position samples per second
- Resonator frequency tuning range ±8%
- ABN cycles/rev configurable up to 16384
- Tested to 66,000rpm (at 1000 AB cycles/rev)
- AB Edge Rate up to 4.5MHz (4170rpm at 16 bits)

Product identification	
Part no.	**Description**
CAM622UE	28-pin UQFN -40°C to 125°C

Figure 1 CAM622UE image

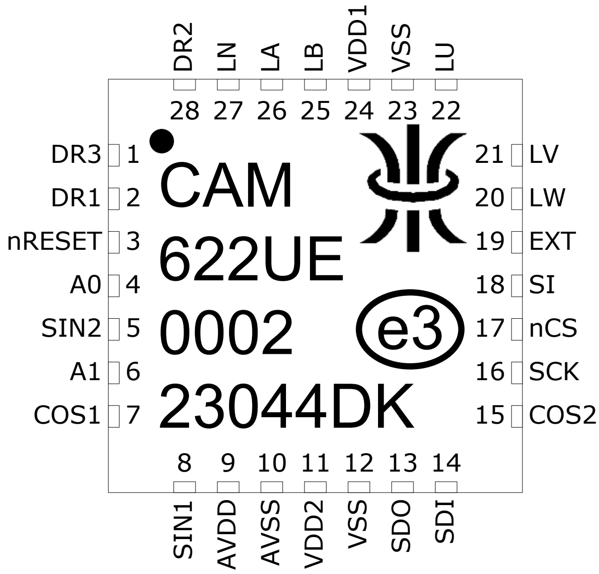


Figure 2 CAM622UE 28-pin SSOP pinout

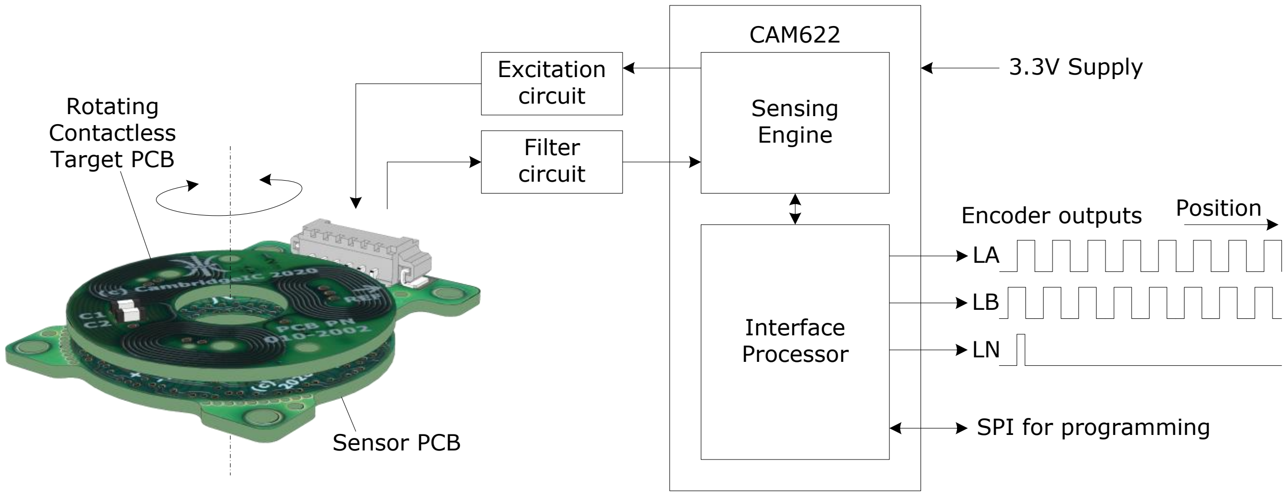


Figure 3 CAM622 function overview

1 Functional Description

Figure 3 shows an overall block diagram of the CAM622, sensor and external circuitry.

Please refer to section 3 for details of the external circuitry required, including the excitation circuit and filter circuit.

1.1 Type B Sensor Overview

The CAM622 works with Type B Sensors. Please refer to a Type B sensor's datasheet for more details of its operation and performance. Please refer to the Type B Sensor Reference Manual for the principle of operation of Type B sensors, and for sensor integration guidelines.

Sensors and targets are both built from conventional PCBs. The sensor comprises 3 or 5 printed coils, and is connected to CAM622 circuitry. The target is an electrical resonator and comprises one printed coil connected to one or more resonator capacitors. The target has no electrical or mechanical connection to the sensor.

Sensors include an excitation (EX) coil that is used to energise the resonator inside the target.

Type B sensors each have a Subtype. When this is non-zero, this indicates that the sensor includes fine (COSA, SINA) and coarse (COSB, SINB) sensor coils. The Subtype is the number of sinusoidal repeats of the fine coil across 360° mechanical rotation. The coarse coils have one sinusoidal repeat across 360° mechanical rotation.

1.2 Sensing Engine Description

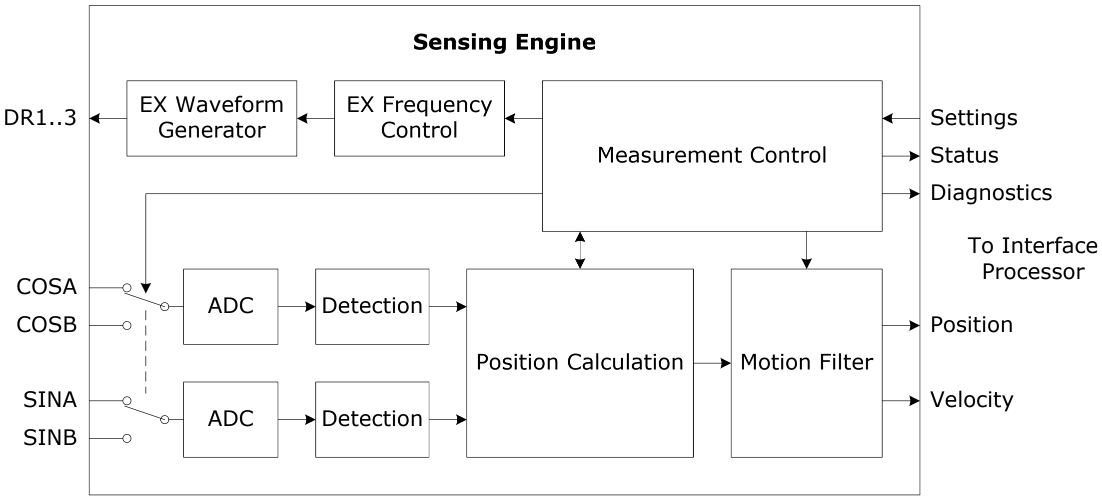


Figure 4 Sensing Engine block diagram

The purpose of the Sensing Engine is to energise the sensor and process its return signals to yield position and velocity data.

The Sensing Engine includes an EX Waveform Generator that outputs the DR1…3 signals used to drive the excitation circuit. It is fully digital and its precise frequency and timing are under the control of an EX Frequency Control block. The target's resonator has a high Q-factor, which means that the frequency of the energising current must closely match its resonant frequency for efficient power transfer. The EX Frequency Control block includes a control loop to maintain a match between the EX Waveform Generator's output frequency and the resonator's frequency.

Signals from the sensor coils are converted to digital signals with ADCs. This allows all remaining processing to be done in the digital domain, for precision and reliability. A multiplexer scheme allows the ADCs to either measure fine sensor coil signals (COSA, SINA) or coarse sensor coil signals (COSB, SINB). A Measurement Control block selects which coils are connected for any given measurement.

Detection blocks process ADC results, and yield data on signal amplitude, frequency and phase. These blocks have a wide but finite frequency range, and this determines the extent to which the Sensing Engine can tolerate resonator frequency variation. This is specified in section 4.

Frequency data from these detection blocks is passed to the EX Frequency Control block, to close the EX frequency control loop and keep the EX frequency matching the resonator frequency.

Signal amplitude and phase data is passed to the Position Calculation block. This manipulates the data in a precise, ratiometric way to yield measurements of the target's position.

Most position measurements are taken using data from the fine coils. Measurements from the coarse coils are only taken at start-up, to first establish absolute position, and again at regular intervals as a cross check. That interval is under the control of the ABSSEL control, see section 6.4. The Position Calculation block combines fine and coarse measurement data to ensure that its output remains absolute across 360° all of the time.

The Sensing Engine's measurement process is extremely rapid, typically repeating with an interval as little as 30µs (see section 6.5).

Position measurements are processed by a Motion Filter. This can reduce measurement noise at the expense of a delayed response to changes in velocity. It also corrects for the small position lag between physical angle and each position update. Please see section 13 for full details of how to configure this filter, and its dynamic performance.

1.3 Interface Processor Description

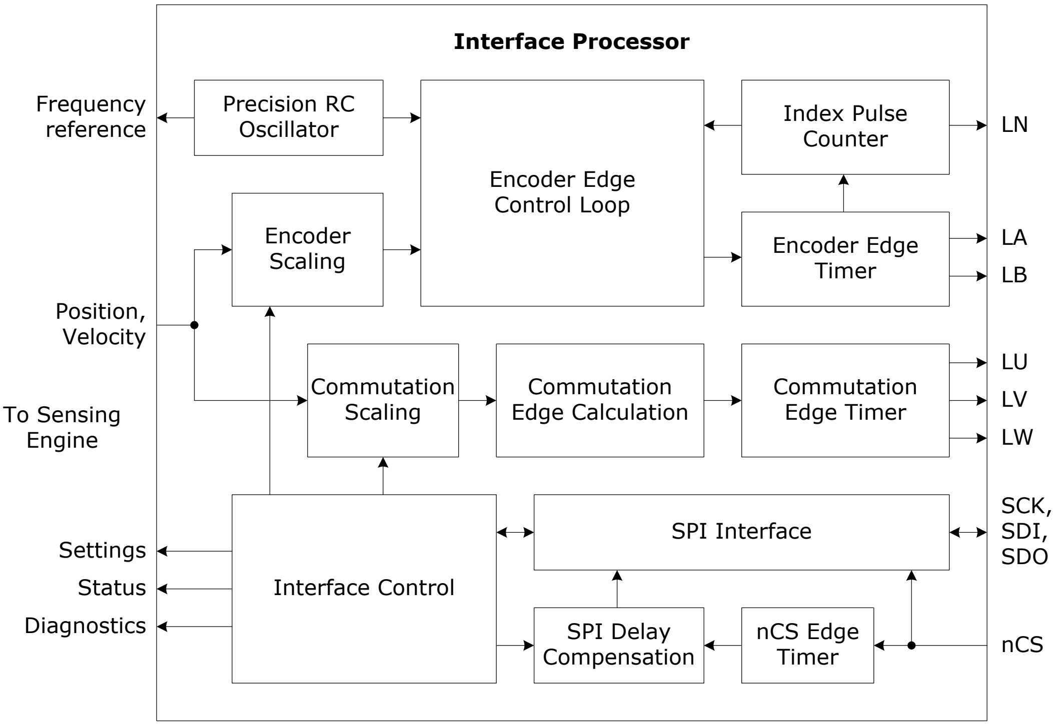


Figure 5 Interface Processor block diagram

The Interface Processor communicates measurement results to the outside world. It can generate ABN signals emulating the outputs of an optical encoder. It can also emulate the output of a BLDC motor's Hall sensors used for commutation at the same time.

Encoder signal settings include the number of AB cycles per revolution and the angular position of the N pulse. Commutation signal settings include the number of pole pairs per revolution and an angle offset. These settings are stored in non-volatile memory. Since they are software settings and not a function of hardware, this allows a single hardware variant to address multiple applications. It allows settings to be configured, adjusted or uploaded in the field under software control.

The Interface Processor also includes an SPI interface. This may be used as a primary interface to a host device in place of encoder and commutation signals. Alternatively it may be used to program the CAM622's non-volatile memory to configure sensing, encoder and commutation settings.

When operating as the primary interface, the exact time of the start of each SPI transaction (nCS low) is measured with a timer. This is used to correct for variable delays between Sensing Engine results and SPI, so that position values reported over SPI reflect the exact position when nCS went low.

The Interface Processor shares a Precision RC oscillator frequency reference with the Sensing Engine. Its frequency tolerance affects the scaling of velocity measurements reported by the Sensing Engine and over SPI. However this tolerance does not affect the frequency of encoder signals. If the Precision RC oscillator's frequency is high then the velocity measurement will be low, and this will be exactly compensated by a higher ABN and UVW edge frequency.

1.4 Pin Functions

Table 1 summarises CAM622 pin functions and type. Please refer to section 2 for electrical characteristics for each type. Signal directions are given from the CAM622 chip's perspective.

Table 1 CAM622 pin names and functions

Signal Name	Type	Function
VDD1, VDD2	Power	Positive supply voltage, internally connected.
AVDD	Analog Input	Analog supply voltage, connect to VDD.
VSS, AVSS	Power	0V connection and common return for sensor inputs.
nRESET	Digital Input	Hardware reset, active low.
nCS	Digital Input	SPI Interface line: Chip Select, active low.
SCK	Digital Input	SPI Interface line: Serial Clock. Serial data is shifted out on the falling edge and captured on the rising edge of SCK.
SDI	Digital Input	SPI Interface line: Serial Data In. Data is captured on the rising edge of SCK.
SDO	Digital and Open Drain Output	SPI Interface line: Serial Data Out. SDO is driven as a digital output by the CAM622 during SPI transactions, when is nCS is low (active). When nCS is high it is Open Drain to allow other slave devices to share the SPI bus.
SI	Digital or open drain output	Sample indicator indicating new results or for controlling an LED.
LA, LB, LN	Digital Outputs	Encoder outputs, single ended, logic level. Use an external line driver to generate RS-422 level ABN signals from these, see section 9.5. Also used as control signals for BiSS operation, see section 10.3
LU, LV, LW	Digital Outputs	Motor commutation outputs, single ended, logic level. Use external MOSFETs to generate open drain UVW signals from these, see section 12.2. Also used as control signals for BiSS operation, see section 10.3
EXT	Digital Output	Configurable function
DR1 – DR3	Digital Outputs	Used to drive external MOSFETs for powering the excitation coil of the resonant inductive position sensor.
COS1, COS2 SIN1, SIN2	Analog Inputs	Used to sense the sensor coil outputs of resonant inductive sensors.
A0, A1	Digital Outputs	Configuration outputs, see section 6.8.
Exposed Pad	Shield	Centre Pad under package, connect to VSS.

2 Electrical Characteristics

2.1 Operating Characteristics

Table 2 Operating characteristics

Item	Min	Max	Comments
Operating Supply Voltage VS	3.1V	3.60V	VDD, AVDD must be greater than 3.0V
Operating Temperature (ambient)	-40°C	125°C	
VS start voltage relative to VSS		0V	For reliable power on reset, and to avoid more than 0.3V difference between VDD and AVDD
VS rise rate relative to VSS	1V/ms	40V/ms	

2.2 Absolute Maximum Ratings

Table 3 Absolute maximum ratings

Item	Max
Voltage between VDD or AVDD and VSS	-0.3V to +4.0V
Voltage on any other pin relative to VSS	-0.3V to (VDD+0.3V)
Current into or out of Digital Output	4mA

2.3 Digital Input Specifications

Table 4 Digital input specifications

Item	Min	Max
Input Low	VSS	0.2 x VDD
Input High	0.8 x VDD	VDD
Input leakage current		±2μA

2.4 Digital Output Specifications

Table 5 Digital output specifications

Item	Min	Max	Comments
Output Low Voltage		0.4V	IOL = 4mA
Output High Voltage	2.4V		VDD=3.3V IOH = -4mA

2.5 Application Memory Characteristics

Table 6 Application Memory characteristics

Item	Min	Max	Comments
Number of non-volatile memory updates		5000	Across Operating Supply Voltage and Operating Temperature
Characteristic retention, -40°C to +125°C	20 years		

2.6 Hardware Reset

The CAM622 chip will be reset when the host device pulls nRST low. The timing parameter TnRSTL is the low time for the nRST signal, and this is specified in Table 7.

Table 7 Reset timing

Parameter	Description	Min
TnRSTL	Duration of nRESET pin low to reliably reset CAM622	2.5 μs

3 External Circuitry

Sections 3.1 and 3.2 detail the circuitry for processing sensor signals using the CAM622. This is all that is needed when the host communicates with the CAM622 chip over SPI.

3.1 Schematic, Sensing Circuitry

Figure 6 is a schematic for circuitry required around the CAM622 chip for its sensing function.

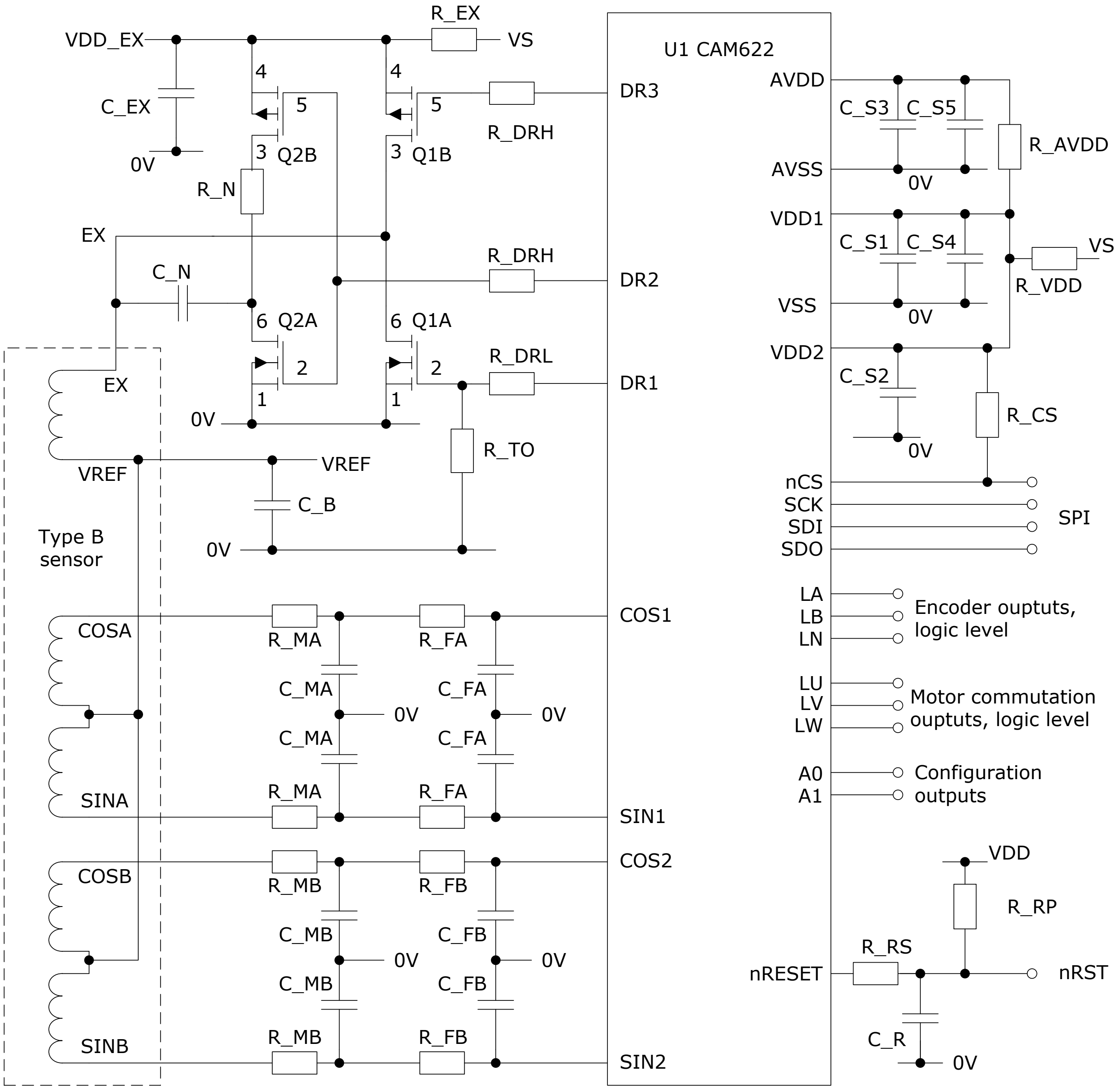


Figure 6 CAM622 external circuitry schematic

Sensors with a Subtype of 0 only have one pair of sensor coils COS and SIN, which are connected to the COSA and SINA nets. They do not use COSB and SINB connections. In this case R_MB, C_MB, R_FB and C_FB may be omitted.

3.2 Components Required, Sensing Circuitry

Table 8 lists the component values required for the schematic of Figure 6.

Table 8 Components required

Circuit Ref	Value, Type	Tolerance		Number required
		Grade A	Grade B	
R_RP, R_CS	10kΩ	±5%		2
R_RS	470 Ω	±5%		1
R_VDD, R_AVDD	0.68Ω	±5%		2
R_EX	10Ω	±5%		1
R_DRL, R_N	220Ω	±5%		2
R_DRH	120Ω	±5%		2
R_TO	1MΩ	±5%		1
R_MA, R_FA	150Ω	±0.1%	±1%	4
R_MB, R_FB (1)	150Ω	±1%		2
C_S1, C_S2, C_S3	100nF	±10%		3
C_S4	22µF, ESR < 0.05 Ω	±20%		1
C_S5, C_EX	10µF, ESR < 0.05 Ω	±20%		2
C_R	1nF	±10%		1
C_N	3.3nF NPO/COG	±5%		1
C_MA, C_FA	220pF NPO/COG	±1%	±5%	4
C_MB, C_FB (1)	220pF NPO/COG	±5%		4
C_B	1µF, ESR < 0.1 Ω	±10%		1
Q1, Q2	Si1016CX			2
U1	CAM622			1

Note(1): May be omitted when the sensor's Subtype is 0.

Table 9 illustrates the effect of component grade choice on reproducibility error, see also section 3.6.

Table 9 Reproducibility error due to filter components, 360°rotary sensor

Subtype	Grade A	Grade B
0	±0.14°	±0.72°
3	±0.05°	±0.24°
5	±0.03°	±0.14°
7	±0.02°	±0.1°
9	±0.016°	±0.08°

3.3 Supply and Reset Circuitry

The circuitry is powered by the VS connection.

R_VDD and C_S4 form an RC filter, which limits the amount of supply ripple on VS that reaches the CAM622's VDD pin. R_AVDD and C_S5 form another RC filter which further reduces ripple on the AVDD. It is essential that the voltage magnitude between the VDD and AVDD pins remains below 0.3V, including when supply voltage is applied to and removed from VS. This condition is met with the component values specified, and providing the maximum VS rise rate specified in Table 2 is respected.

Decoupling capacitors C_S1, C_S2 and C_S3 must be positioned immediately next to the CAM622 chip with short connections to it. Please see Figure 9 for the recommended layout and wiring of these components.

Reset circuitry connects to the CAM622 chip's nRESET pin. C_R and R_RP ensure nRESET is initially low and then high if nRST is not connected to a host device, to ensure the part resets when power is first applied. nRST is an optional connection to the host device. It can simplify updates to internal software over SPI.

3.4 Excitation Circuitry

The sensor is powered by excitation current into its excitation connection EX. Excitation comprises a few cycles of alternating current. This flows between the start and end of excitation, marked EX_START and EX_END in the waveform drawing of Figure 7.

Excitation current has period T_PER marked in the waveforms of Figure 7, and hence frequency 1/T_PER. The CAM622 chip adjusts this period so that the excitation frequency matches the target's resonant frequency.

Excitation current comes from MOSFET Q1. This includes n-channel and p-channel devices Q1A and Q1B respectively, which form a half bridge driver. Their gates are controlled by the CAM622 with non-overlapping signals DR1 and DR3. There is a time delay T_SL between each MOSFET turning off and the next turning on, and this yields high efficiency.

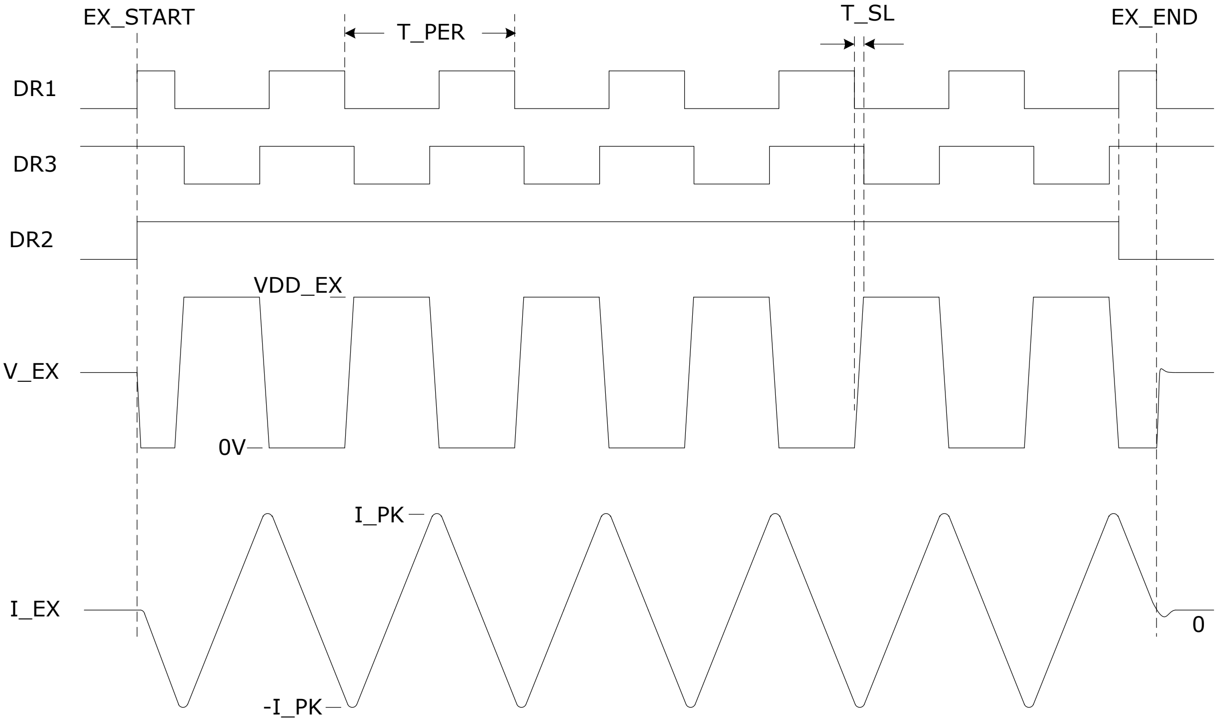


Figure 7 Excitation waveforms

MOSFET Q2A is controlled by DR2 and is on during excitation. This connects capacitor C_N between EX and 0V, so that it conducts excitation coil current when Q1A and Q1B are both off. During this period the excitation current will be approximately constant because the excitation coil is an inductive load. The voltage at EX will therefore slope up or down at a controlled rate. This makes the EX output a trapezoidal waveform approximating a sine wave, to minimise unwanted emissions.

Excitation ends at EX_END, marked in Figure 7, and is followed by resonator detection. At this point Q1A, Q1B and Q2A are all turned off in order to minimise any residual current flowing in the excitation coil during resonator detection, which would otherwise disturb measurements. This is of great benefit to performance, and is one distinguishing feature of resonant inductive sensing compared to conventional (non-resonant) inductive sensing. Conventional inductive sensing requires excitation during detection, so there is no opportunity to switch off the excitation current completely.

MOSFETs are driven by resistors R_DRH and R_DRL. These ensure MOSFETs switch on and off slowly, to avoid unwanted transients during and immediately after excitation.

Resistor R_N helps to absorb any remaining transients in the transition from excitation to detection after EX_END. It is switched into circuit by Q2B, and is also controlled by the DR2.

3.5 Sensor Signal Description

The excitation current I_EX illustrated in Figure 7 is applied to the senor's excitation coil. This couples to the moving target's resonator and causes it to resonate. The resonance builds up while I_EX is applied, and starts to decay when it is removed at EX_END. The decay takes many cycles because the resonator is designed to have a high Q-factor.

The resonator couples back into the sensor's sensor coils. These are patterned so that the coupling between the resonator and sensor coils varies sinusoidally with angle. Coupling to the fine sensor coils COSA and SINA repeats multiple times per target rotation. The number of sinusoidal repeats depends on the sensor design and equals its Subtype. Coupling to the coarse sensor coils COSB and SINB repeats once per rotation. If a sensor's Subtype equals 0 then it only has one pair of sensor coils COS and SIN.

Figure 8 illustrates the end of the excitation current waveform and the resulting resonator signals induced in the fine sensor coils. Coarse sensor signals are similar but are typically smaller.

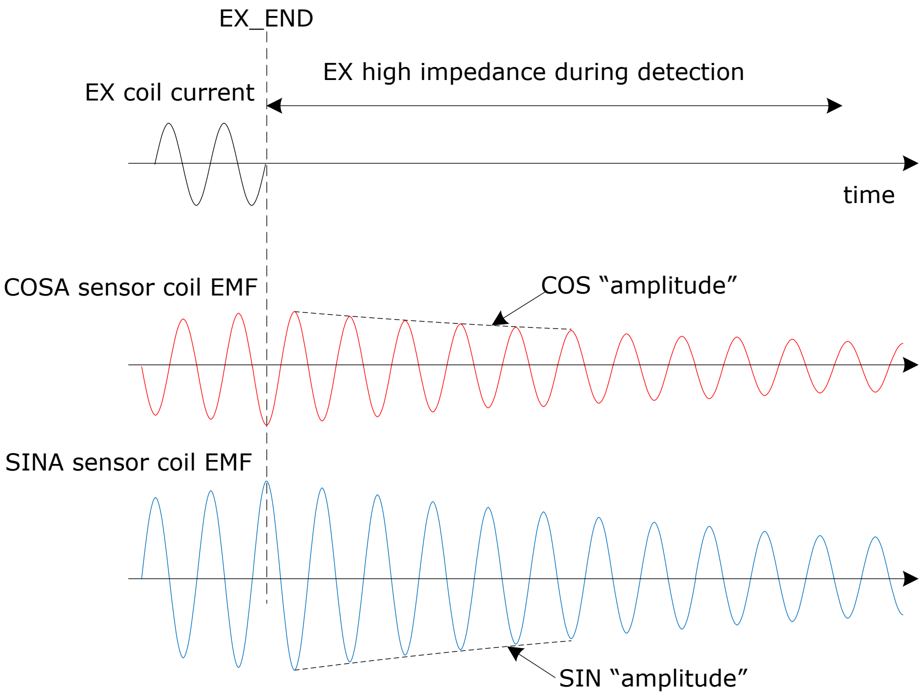


Figure 8 Sensor coil signals COSA and SINA

To determine the angle of the moving target, the CAM622 detects the amplitude of each sensor signal. This is illustrated as the envelope of the waveform in Figure 8 for simplicity. Please refer to the Type B Sensor Reference Manual for more details.

3.6 Sensor Signal Filtering

An external low-pass filter is required to suppress any high-frequency interference coupled into the sensor and its connections. This comprises two stages of RC filtering per sensor signal. This simple approach is ensures gain and phase matching across channels, which is important for accuracy.

Close tolerance filter component values are preferred when accuracy is particularly important, denoted "Grade A" in Table 8. The contribution made by the filter to reproducibility error depends on grade as illustrated in Table 9.

3.7 Interface Circuitry for Other Interfaces

The circuitry described above is required for all interfaces. It is typically sufficient when the CAM622 is interfaced over its SPI interface to a host device on the same PCB.

For applications where encoder outputs are the primary interface it is likely that line drivers will also be required. These boost current and voltage, provide line matching and add differential outputs. Please refer to section 9.5.

Where BiSS is the primary interface, or used to configure ABN operation, the circuitry described in section 10.3 will be required.

Where SENT is the primary interface, signal conditioning options are described in section 11.4.

Motor commutation outputs LU, LV and LW will usually require external MOSFETs to interface to a motor controller, see section 12.2.

3.8 Layout Recommendations

It is recommended to use a PCB with at least 4 layers. A middle layer should include a solid ground plane underneath the CAM622 and the circuitry of Figure 6.

Figure 51 shows the recommended footprint for the CAM622 part. This includes a large centre pad. It is recommended to add vias to this centre pad as shown in the footprint drawing. These are denoted "thermal vias" in Figure 51. They also serve to maintain a low impedance connection between the ground plane and the centre pad. Having a grounded centre pad also allows short and direct connections to the VSS pins, which is important for performance.

Decoupling capacitors C_S1, CS2 and C_S3 and filter capacitors C_FA and C_FB must be positioned immediately adjacent to the CAM622 chip. Their grounded end must be connected directly to the ground plane or an immediately adjacent pin connected to VSS or AVSS.

Figure 9 illustrates the recommended arrangement.

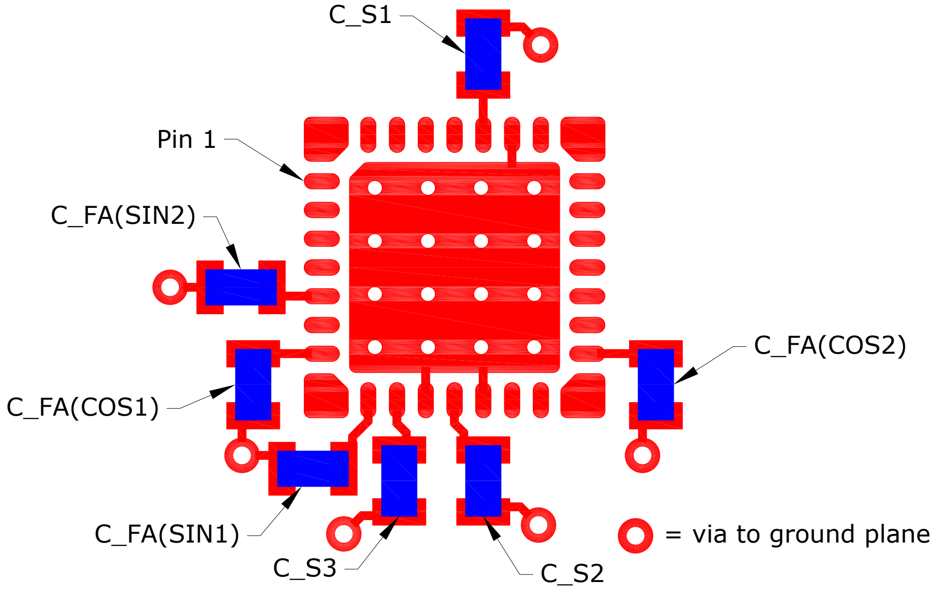


Figure 9 Recommended capacitor locations and wiring

Connections to the CAM622 chip's other pins are less critical.

3.9 Sensor Coil and Filter Network Connections

Figure 10 illustrates connections between sensor coils and the CAM622 circuit, including its filter components. When a sensor coil is connected to the CAM622 circuit, the traces and/or wires forming the connection make a loop. The loop formed by the COSA coil connection is shaded as an example. The loop area for each of the sensor coils must be minimised, in order to minimise coupling to the target and/or any AC magnetic interference.

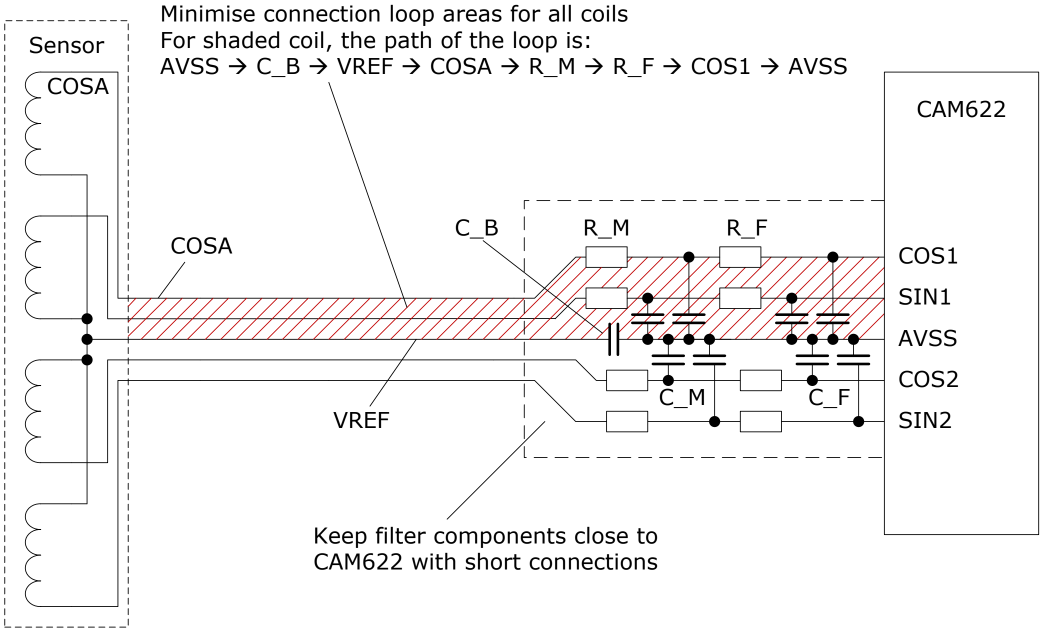


Figure 10 Sensor coil and filter connections to CAM622

Wires used for connection must be run in a tight bundle, on adjacent conductors in a ribbon cable or twisted. Conductors on a PCB should be arranged adjacent to a common VREF conductor as in Figure 11 (a), (b) or (c). These bundles should ideally be less than 100mm long.

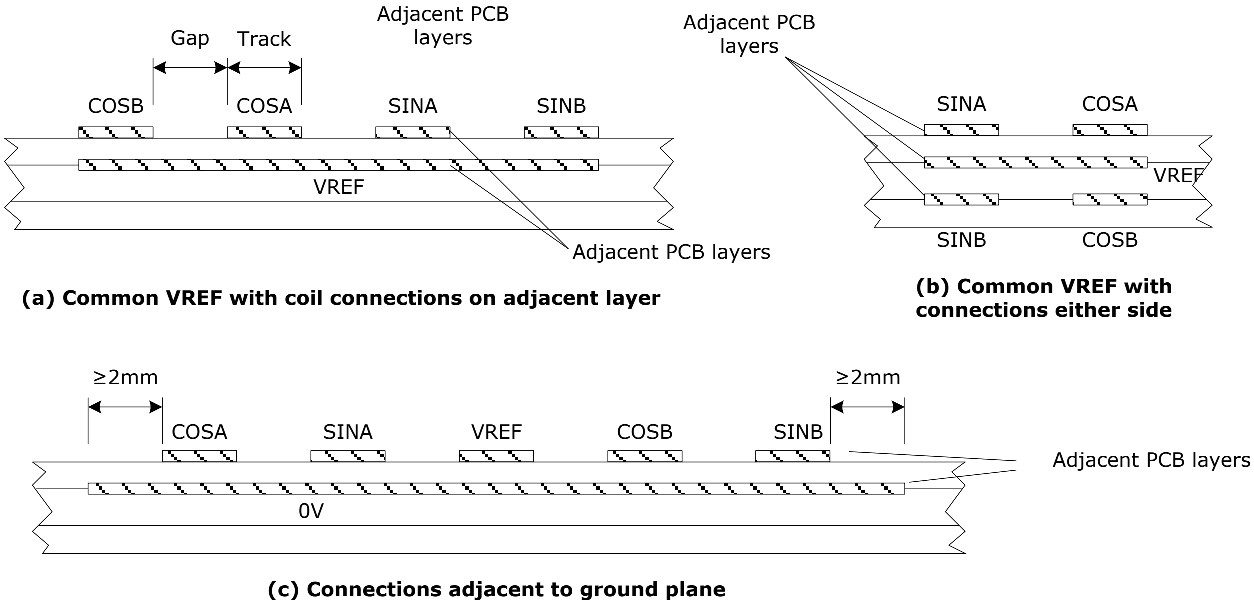


Figure 11 PCB conductor arrangements for sensor coils

Track and Gap figures should be minimised, providing connection resistance does not fall below 2Ω, or below 1Ω for a VREF connection that is common to two or more coils.

3.10 Excitation Circuit and Coil Connections

The CAM622 chip drives the gates of external MOSFETs, which in turn drive current into the excitation coil to energise the resonator inside the target.

To keep the circuit efficient and to minimise emissions, the traces used to form the current paths highlighted in red in Figure 12 must be kept short, direct and as wide as possible. To achieve this, the relevant components (C_EX, Q1, Q2, C_N) must be located together.

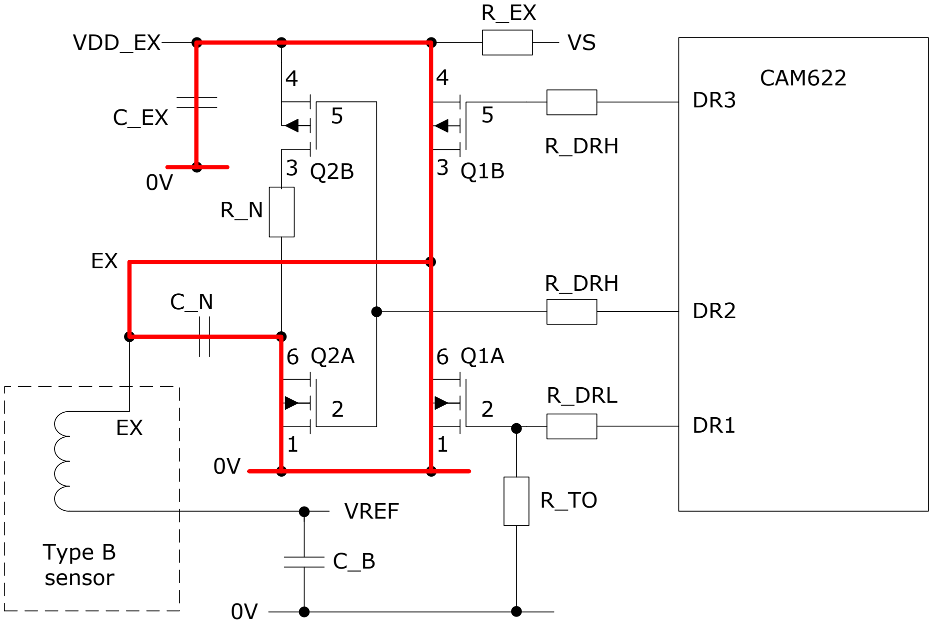


Figure 12 Excitation circuit with critical paths highlighted

Excitation coil connections must be made with a minimum of loop area, in a similar way to sensor coil connections.

The traces connecting the excitation coil (EX and VREF nets) should be at least 0.2mm wide and ideally less than 50mm long.

4 Resonator Detection

Type B sensors detect the position of an electrical resonator inside a moving target. Table 10 lists parameters relevant to the CAM622 chip's detection of resonators, and their values.

Table 10 Resonator Detection Specifications

Parameter	Value	Comment
Nominal Resonator Centre Frequency	833.3kHz	For a reported relative frequency of 0Hz with a nominal CAM622 chip
CAM622 Frequency tolerance across parts and Operating Temperature	±3%	Tolerance of the CAM622's oscillator, and hence tolerance of the Nominal Resonator Centre Frequency and Sample Interval. See section 6.5.
Resonator tuning range Relative to Nominal Resonator Centre Frequency	±8%	For the CAM622 to report VALID Across Operating Temperature Range -40°C to +125°C
Recommended resonator tuning range Relative to Nominal Resonator Centre Frequency	±5%	For best Amplitude and hence resolution
Minimum Resonator Q-factor	12	For the CAM622 to report VALID
Minimum Amplitude (AMPA, Table 13)	3000	Recommended minimum for design purposes
	1000	Absolute minimum to report VALID

Please refer to the Type B Sensor Reference Manual for details, including how to tune the target's resonator to match the CAM622 Resonant Inductive Encoder IC.

5 SPI Hardware

5.1 Overview

This section describes how the CAM622 and a host exchange data over an SPI interface.

5.2 Data Transfer Method

The CAM622 chip is an SPI peripheral. The host controls the nCS, SCK and SDI inputs to the CAM622. The host starts a data transfer by driving nCS low. It sends data to the CAM622 with the SDI line, and provides the CAM622 with a serial clock line SCK. The CAM622 detects each SDI bit on the rising edge of SCK. The CAM622 sends data back to the host with the SDO line. SDO changes state on the falling edge of SCK. The host should detect the state of SDO on each rising edge. This is commonly referred to as SPI Mode 0. The beginning and end of an SPI transaction are illustrated in Figure 13.

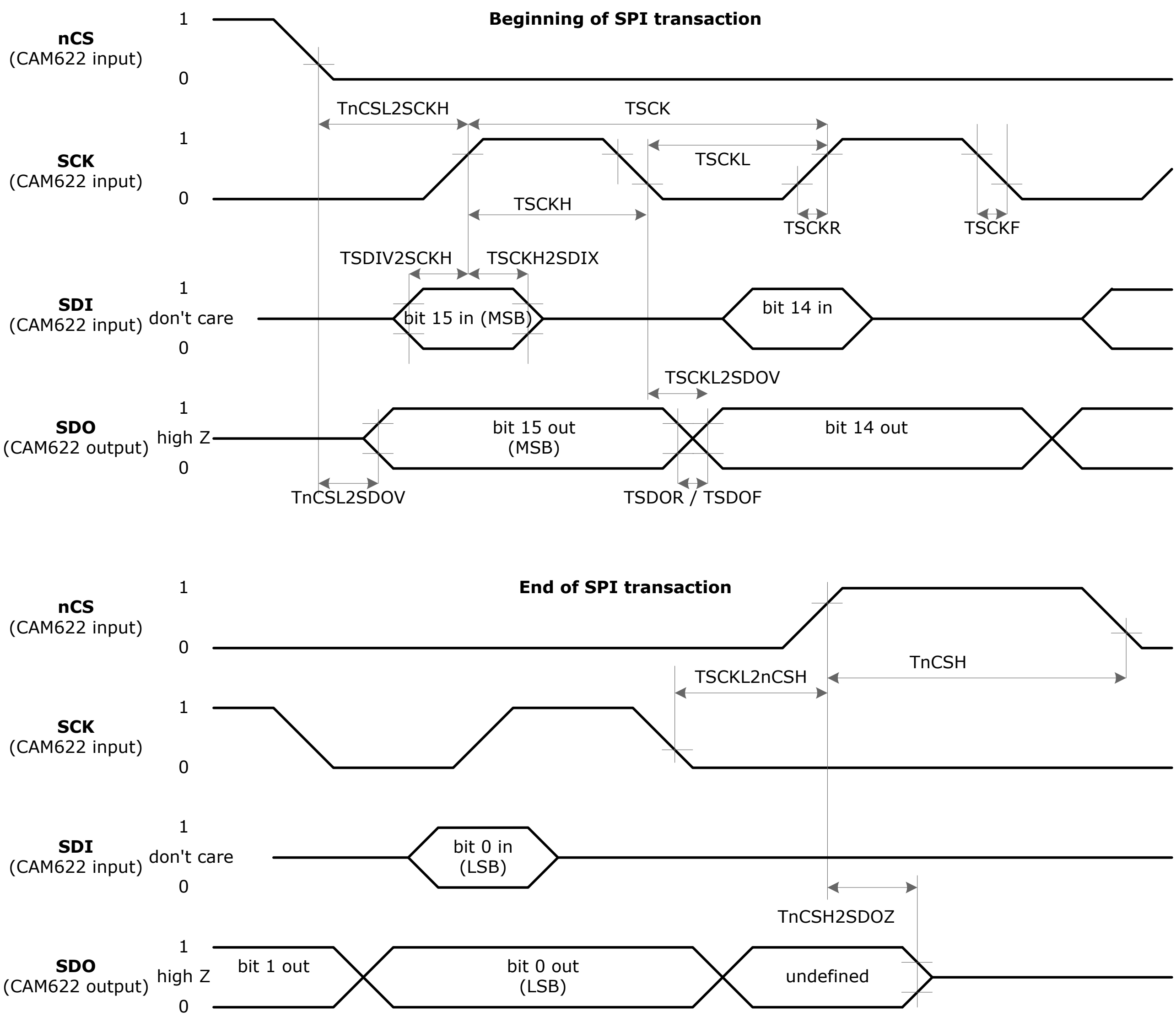


Figure 13 SPI Data Transfer

The timing parameter marked TnCSH is the time that nCS remains high between successive SPI transactions. Its minimum value depends on the details of the preceding SPI transaction. Several values are given in Table 11. For example TnCSH_read is the time between read operations when ABEN=0, as illustrated in Figure 18.

Data is transmitted most significant bit (MSB) first. The CAM622 interprets data it receives on SDI as 16-bit words. This data is denoted WIN0, WIN1, WIN2 in Figure 14. The host should also interpret data output by the CAM622 as 16-bit words, denoted WOUT0, WOUT2, WOUT2.

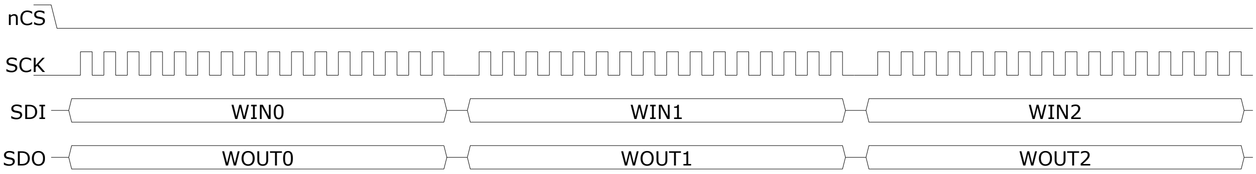


Figure 14 Arrangement of SPI words

All SPI transactions MUST be bounded by chip select (nCS) being driven low at the start and being driven high at the end. The SPI interface will not function if nCS remains permanently low.

5.3 SPI Following Reset

The CAM622 includes power on reset circuitry that holds it in reset until its power supply has reached a level suitable for normal operation. It may also be reset by pulling nRESET low, or by setting the RESET bit (section 6.8).

Once out of reset, the CAM622 performs an internal self test and validity checking procedure. This process must complete for the CAM622 to subsequently enter its normal operating mode, to allow measurements and register access over SPI. To complete successfully, the host must maintain nCS high during the procedure. The minimum high time TR2nCSL(min) between coming out of reset and nCS low is specified in Table 11.

If the host interrupts self test and validity checking by pulling nCS low, the CAM622 will enter its bootloader mode. Please refer to document "Uploading Application Code" for details of this process.

5.4 SPI Transaction Timings

Table 11 specifies the timing parameters required for correct operation of the CAM622 SPI interface.

Table 11 SPI Transaction Timings

Parameter	Description	Min	Max	Units
TR2nCSL	Time between coming out of reset and first nCS low	20	-	ms
TSCKL	SCK Input Low Time	30	-	ns
TSCKH	SCK Input High Time	30	-	ns
TSCK	SCK clock period	66	-	ns
TSCKR	SCK Input Rise Time	-	7.5	ns
TSCKF	SCK Input Fall Time	-	7.5	ns
TSDOR	SDO Rise Time (50pF load)	-	10	ns
TSDOF	SDO Fall Time (50pF load)	-	10	ns
TSDIV2SCKH	SDI Setup Time	30	-	ns
TSCKH2SDIX	SDI Hold Time	30	-	ns
TnCSL2SDOV	First SDO state valid after nCS low edge	-	50	ns
TSCKL2SDOV	SDO state valid after SCK low edge	-	20	ns
TnCSL2SCKH	nCS low to SCK edge	500	-	ns
TSCKL2nCSH	Last SCK edge to nCS high	90	-	ns
TnCSH2SDOZ	nCS high to SDO high Z	-	50	ns
TnCSH_config	Write-read intended configuration up to and including SICONFIG register	20	-	µs
TnCSH_read	Read from any registers, when ABEN=0	4.0	-	µs
TnCSH_ABNread	Read from any registers, when ABEN=1	20	-	µs
TnCSH_START	Write-read CTRL register only, following a clearing of START bit with ABEN=1	20	-	µs
	Write-read CTRL register only, other	10	-	µs
TnCSH_NVconfig	Write-read standalone configuration and save to non-volatile memory.	50	-	ms

6 Register Access and Descriptions

The CAM622 includes a set of registers used to configure its operation. A host may also read version information and the results of measurements from registers.

Registers may be accessed over SPI, see section 6.1. Once configured for BiSS operation, they may also be accessed over BiSS, see sections 10.9 and 10.10.

6.1 Register Access over SPI

The host accesses CAM622 registers over the SPI interface, see section 5. Each SPI transaction, the host sends a number of data words to the CAM622 denoted WIN0, WIN1, WIN2… and the CAM622 returns data words to the host denoted WOUT0, WOUT1, WOUT2…

To read registers without modifying their contents, the host must clear the first bit of the first word it sends to the CAM622 (WnR=0, bit 15 of WIN0). The contents of the CAM622's registers are returned in the WOUT words. WOUT0 corresponds to the contents of register address 0x00 (CTRL), WOUT1 corresponds to the contents of register address 0x01 (POSH), and so on. Please see Figure 15. Apart from the WnR bit, the CAM622 ignores all of the WIN words it receives from the host.

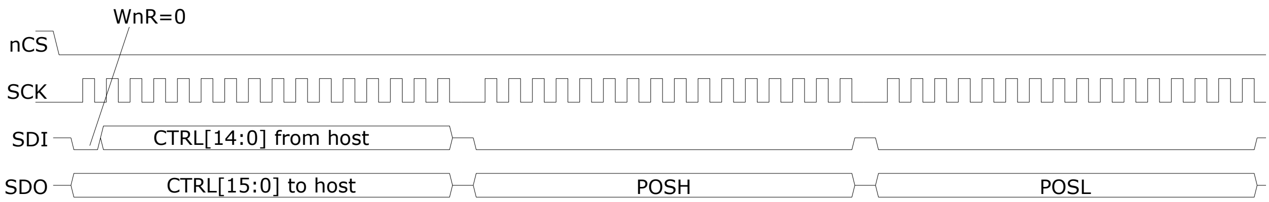


Figure 15 Register read operation

To write new values to registers, the host must set the first bit of the first word it sends to the CAM622 (WnR=1, bit 15 of WIN0). The CAM622's register contents are updated based on the WIN0, WIN1, WIN2… values it receives from the host. WIN0 is written to the register at address 0x00 (CTRL), WIN1 is written to the register at address 0x01, and so on. Please see Figure 16.

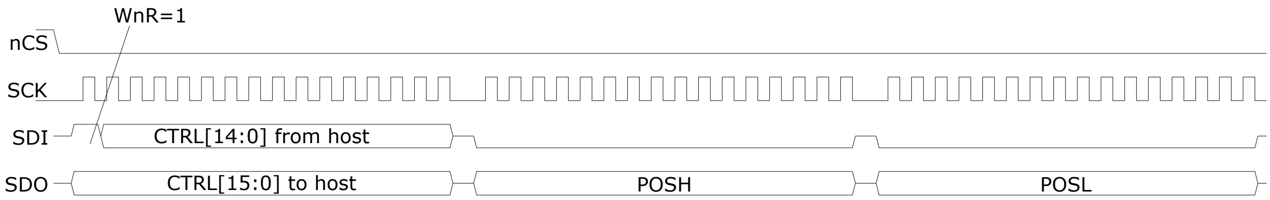


Figure 16 Register write-read operation

During a write operation, the CAM622 returns register contents in WOUT0, WOUT1, WOUT2… Note these are the contents of the registers as they stood before the SPI transaction. An SPI transaction with the WnR bit set therefore performs both write and read operations, and is referred to below as "write-read".

Register access may be either read only ("R") or write-read ("WR"). The contents of read only registers are only updated by the CAM622. When performing a write-read operation, the host must write 0x0000 to read only registers. The contents of write-read registers are updated by the host, using a write-read SPI transaction.

The control register CTRL is an exception, including bits with both read only and write-read access.

Unimplemented registers and unimplemented bits within registers must be written to 0 when the host performs a write-read operation.

6.2 Register Listing

This section provides an overview of registers and their locations. The tables below list register addresses, names, a brief summary of their function and, where necessary, the section number for more details.

Table 12 Control register (includes write-read and read only fields)

Address	Register Name	Function	Section
0x00	CTRL	Various fields for control and status information	6.3

Table 13 Results registers (read only *)

Address	Register Name	Function	Section
0x01	POSH	Position measurement, high word	7.5
0x02	POSL	Position measurement, low word	
0x03	VELH	Velocity measurement, high word	
0x04	VELL	Velocity measurement, low word	
0x05	AMPA	Amplitude measurement, fine coils	
0x06	AMPB	Amplitude measurement, coarse coils	
0x07	BAMISMATCH	Position mismatch between coarse and fine coils	
0x08	FNUM	Resonator frequency measurement	
0x09	FILTLVL	Current value of Motion Filter level setting	
0x0A	CRC	Checksum of registers 0x00 to 0x09 inclusive, not yet implemented	

(*) The POSH register is used as a write-read register when programming a CAM622 for autonomous operation. The value 0x0C1C must be written to POSH in order for the SAVE bit (section 6.8) to take effect. See section 0

The 4 Information Registers may be configured to equal any register values from 0x10 (SECONFIG) onwards.

Table 14 Information registers (read only)

Address	Register Name	Function	Section
0x0B	INFO0	By default equals CAMID (0x0622 for "CAM622")	6.10
0x0C	INFO1	By default equals SYSVER (Version of Application Code)	
0x0D	INFO2	By default equals BOOTVER (Version of Bootloader)	
0x0E	INFO3	By default equals SEVER (Version of Sensing Engine)	
0x0F		Unimplemented	

Table 15 Sensing Engine control registers (write-read)

Address	Register Name	Function	Section
0x10	SECONFIG	Sensing Engine configuration	6.4
0x11	INTERVAL	Interval between measurements	6.5
0x12	FLTLVLS	Maximum and minimum filter levels	6.6
0x13	HYAD	Hysteresis and adaptive filter sensitivity	6.7
0x14		Unimplemented	
0x15		Unimplemented	
0x16		Unimplemented	

Table 16 System control registers (write-read)

Address	Register Name	Function	Section
0x17	SYSCONFIG	System configuration	6.8
0x18	SICONFIG	Sample indicator control	0
0x19	INDEX10	Controls the contents of INFO0 and INFO1	6.10
0x1A	INDEX32	Controls the contents of INFO2 and INFO3	
0x1B		Unimplemented	

Table 17 Versions registers (read only)

Address	Register Name	Function	Section
0x1C	CAMID	Returns 0x0622 for "CAM622"	
0x1D	SYSVER	Version of Application Code	
0x1E	BOOTVER	Version of Bootloader	
0x1F	SEVER	Version of Sensing Engine	
0x20		Unimplemented	
0x21		Unimplemented	
0x22		Unimplemented	

Table 18 ABN results registers (read only)

Address	Register Name	Function	Section
0x23		Unimplemented	
0x24		Unimplemented	
0x25		Unimplemented	
0x26		Unimplemented	
0x27	ABCOUNTH	Current AB count value, whole counts	
0x28	ABCOUNTL	Current AB count value, fractional part	

Table 19 ABN control registers (write-read)

Address	Register Name	Function	Section
0x29	ABCONFIG	Fields for controlling ABN signals	6.11
0x2A	MAXABFREQ	Controls max AB edge frequency, default is the max value 0x8D (141)	9.8
0x2B	NPOS	Controls position of N signal, default 0x0000	9.6
0x2C	ABCYC	Controls number of AB cycles per revolution, default 0x0000 (16384)	9.6
0x2D		Unimplemented	

Table 20 UVW results registers (read only)

Address	Register Name	Function	Section
0x2E	UVWCOUNTH	Current UVW state number, whole states	
0x2F	UVWCOUNTL	Current UVW state number, fractional states	

Table 21 UVW control registers (write-read)

Address	Register Name	Function	Section
0x30	UVWCONFIG	Fields for controlling UVW signals	6.11
0x31	UVWPOS	Controls start position of UVW signals	
0x32	UVWCYC	Controls number of cycles (pole pairs) of UVW signals	

Table 22 BiSS control registers (write-read)

Address	Register Name	Function	Section
0x36	BISSCONFIG		6.13
0x38	BISSKEY1	Used to secure settings against unauthorised access over BiSS.	
0x39	BISSKEY0		
0x3A	BISSMID	Controls the Manufacturer's ID reported over BiSS	Table 45
0x3B	BISSDID2	Controls the Device ID reported over BiSS	
0x3C	BISSDID1		
0x3D	BISSDID0		
0x3E	BISSSNUM1	Controls the Serial Number reported over BiSS	
0x3F	BISSSNUM0		

Table 23 SENT control registers (write-read)

Address	Register Name	Function	Section
0x41	SENTCONFIG1	Controls SENT signal behaviour and data format	6.15
0x42	SENTCONFIG2	Controls SENT timings	
0x44	SENTMSG1	Fixed values returned in diagnostic data that a customer can use for version numbers and serial numbers.	Table 56
0x45	SENTMSG2		
0x46	SENTMSG3		

6.3 CTRL Register

The host accesses CTRL with every SPI transaction, because it located at address 0x00 corresponding to the first word of the SPI transaction. Unlike all other registers, some of the bits that the host accesses are different between writes and reads.

Table 24 shows the names of bits within CTRL controlled by data coming from the host on the SDI line (WIN0 inFigure 14). As noted in section 6.1, the top bit (WnR) controls whether the SPI transaction is to read registers (0) or to write-read (1). The contents of subsequent registers (and the START bit) will only be updated if WnR=1. They will remain unaffected if WnR=0.

Table 24 CTRL register, bits sent from host

CTRL	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	WnR	-	-	-	-	-	-	-	-	-	-	-	-	-	-	START
Access	W	-	-	-	-	-	-	-	-	-	-	-	-	-	-	W

Setting the START bit initiates measurements, and clearing it stops measurements. When START=1, the host may only access registers up to and including INFO3 (address 0x0E).

Table 25 shows the names of bits that the host reads from CTRL.

Table 25 CTRL register, bits sent from CAM622

CTRL	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	SYSID[3:0]				COUNT[3:0]				-	-	-	VALID	-	-	-	START
Access	R				R				-	-	-	R	-	-	-	R
Default	0xA				0x0							0				0

SYSID is the System ID. When there are multiple devices on an SPI bus, this may be used to check the correct device is responding. Its value may be changed by the host, see section 6.8. The host may also use SYSID as a basic check that communication with the CAM622 is intact, by checking its value against the expected one.

SYSID may also be used as a check for an unexpected reset. The host should change its value when configuring the CAM622. If the host subsequently reads the default SYSID value instead of this new one, a reset has occurred.

COUNT increments each time a measurement is completed. If the host is designed to read the results of each measurement, for example reading results in response to the signal indicator (section 7.4), then COUNT enables the host to check it has not missed results or read the same set again. COUNT increments in a modulo fashion, so the next value after 0xF is 0x0.

The CAM622 reports VALID=1 when the result of the last measurement was valid. This means the target is in range of sensor, its resonator frequency is within the CAM622 resonator tuning range (section 4) and the Motion Filter has initialised (section 13.1).

6.4 SECONFIG Register

SECONFIG includes fields for configuring the Sensing Engine.

Table 26 SECONFIG register

SECONFIG	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DDC	-	-	-	ABSSEL[3:0]				-	-	SUBTYPE[5:0]					
Access	WR	-	-	-	WR				-	-	WR					
Default	0	-	-	-	0xF				-	-	0x03					

The DDC bit disables delay compensation applied by the Motion Filter, see section 13.1.

ABSSEL configures the number of measurements from the sensor's fine (A) coils between every measurement from its coarse (B) coils, when valid. This number will be denoted ABSCOUNT. Please see section 1.2 for more details. ABSCOUNT is calculated from Equation 1:

Equation 1

$$ABSCOUNT = 2^{ABSSEL}$$

The default value of ABSSEL is 0xF, or 15 in decimal. In this case the value of ABSCOUNT given by Equation 1 is 32768. When operating with the nominal sample interval (see section 6.5) this means a coarse measurement takes place approximately very second.

SUBTYPE is a property of the sensor. Precision sensors that include both fine and coarse coils have a SUBTYPE value greater than 1. SUBTYPE is the number of fine COS/SIN periods per coarse COS/SIN period. Basic sensors only have one COS/SIN pair of sensor coils, and their SUBTYPE value is 0. Please refer to the Type B Sensor Reference Manual for more details SUBTYPE.

A precision sensor's SUBTYPE value is included in its full name. For example the 25mm B3 Precision Through-Hole Sensor has SUBTYPE=3 (the number after "B"). If in doubt, please refer to the sensor's datasheet. Position measurements reported by the CAM622 will always be wrong unless SUBTYE is configured correctly for the connected sensor.

6.5 INTERVAL Register

INTERVAL controls the interval between the start times of successive Sensing Engine measurements.

Table 27 INTERVAL Register

INTERVAL	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	INTERVAL[15:0]															
Access	WR															
Default	0x012C															

INTERVAL is nominally in multiples of 0.1µs, so that the relationship between sample interval in µs and INTERVAL is given by Equation 2:

Equation 2

$$Sample\ Interval/\mu s = INTERVAL\ \times 0.1\mu s$$

For example, INTERVAL is 0x012C by default which is 300 decimal, so Sample Interval is 300 x 0.1µs = 30µs. INTERVAL must be set to this default value for ABN encoder outputs to function correctly.

These figures are subject to a tolerance associated with the CAM622's on-chip oscillator. This tolerance is specified in Table 10 ("CAM622 Frequency tolerance across parts and Operating Temperature").

6.6 FLTLVLS Register

FLTLVLS includes the filter level controls MAXFILTLVL and MINFILTLVL, arranged as in Table 28. Please refer to section 13 for details on how these are used.

Table 28 FLTLVLS Register

FLTLVLS	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	MAXFILTLVL[7:0]								MINFILTLVL[7:0]							
Access	WR								WR							
Default	0xFF								0x32							

6.7 HYAD Register

HYAD includes the HYSESET control which configures the amount of hysteresis added to position measurements when they are used to synthesise ABN encoder and UVW motor commutation signals. Please see section 9.7

HYAD also includes ADAPTSENS, which controls the sensitivity of adaptive filtering. Please see section 13.5.

These registers are arranged as shown in Table 29.

Table 29 HYAD Register

HYAD	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	HYSTSET[7:0]								ADAPTSENS[7:0]							
Access	WR								WR							
Default	0x50								0x82							

6.8 SYSCONFIG Register

SYSCONFIG includes several fields and bits that control system operation, and its arrangement is shown in Table 30.

Table 30 SYSCONFIG Register

SYSCONFIG	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	SYSID[3:0]				SAVE	FACTORY	BOOT	RESET	A1 CTRL	A0 CTRL	A1	A0	-	EXTMODE[2:0]		
Access (SPI)	WR				WR	WR	WR	WR	WR	WR	WR	WR	-	WR		
Access (BiSS)	WR				WR	WR	WR	WR	R	R	R	R	-	WR		
Default	0xA				0	0	0	0	-	-	1	1	-	2		

SYSID is the System ID, whose value is read out of the CAM622 in the first word of each SPI transaction, see section 6.3. The host may change its value by writing a new value to SYSCONFIG.

The host uses the SAVE bit to signal to the CAM622 that register values should be written to non-volatile memory in preparation for standalone operation. It uses the FACTORY bit to return non-volatile memory to factory defaults. Please see section 8. When writing to SYSCONFIG for other reasons, the host must write 0 to SAVE and FACTORY.

Setting the BOOT bit puts the CAM622 into its bootloader mode in preparation for uploading new Application Code. Please refer to document "Uploading Application Code" for details of this process. When writing to SYSCONFIG for other reasons, the host must write 0 to BOOT.

When its RESET bit is set, the CAM622 performs a software reset following the SPI transaction. When writing to SYSCONFIG for other reasons, the host must write 0 to RESET.

The A0 and A1 bits control the states of the CAM622's A0 and A1 output pins. These are typically used to configure external devices. When set, A0CTRL and A1CTRL give control of A0 and A1 respectively to the CAM622, to support configurations including that of Figure 33 where the CAM622 controls A0 and/or A1 dynamically. The A0CTRL and A1CTRL bits can only be written over SPI, not BiSS. This is to prevent a host from changing settings over BiSS that might render BiSS unusable, potentially rendering the product unusable. The A0 setting can only be modified over BiSS if A0CTRL=0, and The A1 setting can only be modified over BiSS if A1CTRL=0. This is also to prevent a host from changing settings over BiSS that might render BiSS unusable.

EXTMODE controls the function of the EXT pin, as summarised in Table 31.

Table 31 EXTMODE Function

EXTMODE[2:0]	**Function**
0	Inactive (EXT = NOT EXTAH)
1	LED control
2	Activates on VALID measurement
3	Reflects rotation direction when ABEN=1 and VALID
4 – 7	Reserved

When set to 1, EXT controls an LED, to provide a visible indication of status. EXT will activate when the CAM622 measurements are VALID with healthy signal level. It will toggle on and off repeatedly to indicate VALID measurements with low signal level. It will inactivate when measurements are not VALID. LEDTHRESHOLD configures the threshold between low and healthy signal level, and it is in units of reported amplitude (AMPA). Please refer to a sensor's datasheet for expected amplitude values. EXTAH controls whether EXT is active high (EXTAH=1) or active low (EXTAH=0), except when EXTMODE=3 when EXTAH is ignored. LEDTHRESHOLD and EXTAH are both fields in the SICONFIG register, see section 6.9 below.

6.9 SICONFIG Register

SICONFIG controls the behaviour of the CAM622's SI and EXT outputs. It includes the fields shown in Table 32.

Table 32 SICONFIG register

SICONFIG	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LEDTHRESHOLD[11:0]												EXTAH	SIAH	SIDO	-
Access	WR												WR	WR	WR	-
Default	0x0FA0 (decimal 4000)												0	0	0	-

SI is a sample indicator output pin. It activates to indicate when each measurement completes, and new data is available from the Results Registers. A host may synchronise its SPI transactions to SI in order to minimise latency between actual position and obtaining measurements, see section 7.4. The SIAH bit configures whether SI is active high (SIAH=1) or active low (SIAH=0). The SIDO bit configures whether SI is a digital output (SIDO=1) or open drain (SIDO=0).

The functions of LEDTHRESHOLD and EXTAH are detailed in section 6.8 above.

6.10 INDEX10 and INDEX32 Registers

The 4 Information Registers (Table 14) are arranged immediately after the Results Registers. Their contents are configurable under host control, using the INDEX10 and INDEX32 registers. The Information Registers are arranged immediately after the Results Registers. An SPI host can read out Results Registers and Information Register values when the CAM622's ABN encoder outputs are enabled (ABEN=1). Indexing therefore allows a host to read selected register values faster than normal, and without the normal restriction preventing registers beyond address 0x10 from being accessed when ABEN=1.

Table 33 INDEX10 Register

INDEX10	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	INDEX1[6:0]							-	INDEX0[6:0]						
Access	-	WR							-	WR						
Default	0	0x0D							0	0x0C						

Table 34 INDEX32 Register

INDEX32	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	-	INDEX3[6:0]							-	INDEX2[6:0]						
Access	-	WR							-	WR						
Default	0	0x0F							0	0x0E						

The INDEX0 register controls the contents of INFO0 at address 0x0B. The INDEX1 register controls the contents of INFO1 at address 0x0C. The INDEX2 register controls the contents of INFO2 at address 0x0D. The INDEX3 register controls the contents of INFO3 at address 0x0E. In each case, the register value appearing in the respective INFO register is equal to the INDEX value plus 0x10.

The default value for INDEX0 shown in Table 33 makes INFO0 equal to 0x0C + 0x10 = 0x1C, which is the address of the CAMID register. This therefore appears in INFO0 by default. Similarly INFO1 equals SYSVER, INFO2 equals BOOTVER and INFO3 equals SEVER by default.

6.11 ABCONFIG Register

ABCONFIG configures the CAM622's encoder outputs LA, LB and LN, and includes the bits shown in Table 35.

Table 35 ABCONFIG Register

ABCONFIG	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	ABEN	-	-	-	-	-	-	-	-	-	-	NSTART	ABDIR	ANEG	BNEG	-
Access	WR	-	-	-	-	-	-	-	-	-	-	WR	WR	WR	WR	-
Default	0	-	-	-	-	-	-	-	-	-	-	1	0	0	0	-

Please refer to section 9 for how to configure these bits, including their descriptions listed in Table 43.

Setting the ABEN bit enables the encoder outputs LA, LB and LN. When taking measurements over SPI, ABEN must be 0 for the CAM622 to respond to SPI transactions with a minimum of delay, and to allow Latch Point Compensation (section 13.6).

6.12 UVWCONFIG Register

UVWCONFIG configures the CAM622's motor commutation outputs U, V and W, and includes the bits shown in Table 36.

Table 36 UVWCONFIG Register

UVWCONFIG	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	UVWEN	-	-	-	-	-	-	-	-	-	-	-	UVWDIR	-	-	2PH
Access	WR	-	-	-	-	-	-	-	-	-	-	-	WR	-	-	WR
Default	0	-	-	-	-	-	-	-	-	-	-	-	0	-	-	0

Please refer to section 12 for how to configure these bits.

Setting the UVWEN bit enables the motor commutation outputs U, V and W. When taking measurements over SPI, UVWEN must be 0 for the CAM622 to respond to SPI transactions with a minimum of delay, and to allow Latch Point Compensation (section 13.6).

6.13 BISSCONFIG Register

BISSCONFIG configures the CAM622 for communication over an optional BiSS interface.

Table 37 BISSCONFIG Register

BISSCONFIG	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	BISSEN	ESE	CB_SLO	-	-	-	-	EDS	-	-	-	-	BISS FORMAT[3:0]			
Access (SPI)	RW	RW	RW	-	-	-	-	RW	-	-	-	-	RW			
Access (BiSS)	R	R	R	-	-	-	-	RW	-	-	-	-	RW			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0x0			

Please refer to Table 45 for how to configure these bits.

The BISSEN, ESE and CB_SLO bits can only be written over SPI, not BiSS. This is to prevent a host from changing settings over BiSS that might render BiSS unusable, potentially rendering the product unusable.

6.14 BISSKEY Registers

The BISSKEY0 and BISSKEY1 registers are used to secure settings against unauthorised access over BiSS.

Table 38 BISSKEY1 Register

BISSKEY1	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	BISSKEYCODE[7:0]								BISSKEY[23:16]							
Access	WR								WR							
Default	0x00								0x00							

BISSKEY is a 24-bit field formed by combining the BISSKEY[24:16] bits in the BISSKEY1 register with the BISSKEY[15:0] bits in the BISSKEY0 register.

Please refer to Table 45 for how to configure KEYCODE and BISSKEY.

6.15 SENTCONFIG Registers

The SENTCOFIG1 and SENTCOFIG2 registers are used to configure the CAM622's SENT interface, which is detailed in section 11. Please refer to Table 56 for the functions of these registers.

Table 39 SENTCONFIG1

SENTCONFIG1	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	SENT EN	SENT POL	SENT OD	-	-	-	-	-	-	-	-	-	SENT FORMAT[3:0]			
Access	RW	RW	RW	-	-	-	-	-	-	-	-	-	RW			
Default	0	0	0	0	0	0	0	0	0	0	0	0	0x0			

Table 40 SENTCONFIG2

SENTCONFIG2	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Field	SENTTICK[7:0]								SENTCOUNT[7:0]							
Access	RW								RW							
Default	0x1E								0x21							

7 Reading Measurements Over SPI

This section explains how to connect a host to the CAM622 when SPI will be the primary means of communication. Please refer to section 8 for using SPI to configure the CAM622 for standalone operation.

7.1 Host Connections to CAM622 Circuitry

A host connects to the CAM622 circuitry of Figure 6 using the 4 SPI lines nCS, SCK, SDI and SDO, as illustrated in Figure 17. Host control over nRST is also strongly recommended, so that the host can put the CAM622 into a known state prior to operation.

An nRST connection also simplifies bootloader operation. Please refer to document "Updating Application Code" for details of how to program Application Code using the CAM622's bootloader.

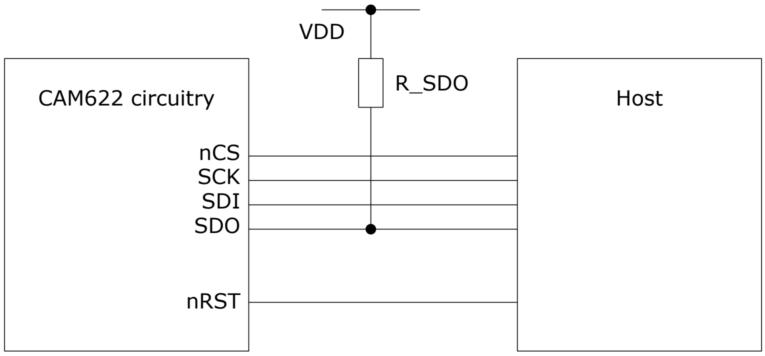


Figure 17 Host connections to CAM622 circuitry

The CAM622 drives SDO high and low when nCS is low (SPI transaction in progress). When nCS is high, the CAM622 makes its SDO output a high impedance. This allows other devices to share the same SPI bus. To avoid SDO floating, a pull-up resistor is needed on SDO. If this can not be provided within the host device itself, it is recommended to add one: R_SDO illustrated in Figure 17.

7.2 Factory Default Register Settings

This section assumes the CAM622 chip has not been configured for standalone operation. If it has, this is likely to interfere with taking measurements over SPI, because the CAM622 will contain non-default register values saved to non-volatile memory. For example, the ABEN bit is usually set for standalone operation, and this will interfere with SPI communication.

To avoid issues such as these, a CAM622 IC that was previously used for standalone operation should have its factory defaults restored, see section 8.5. Note that this operation should only be performed once or a small and limited number of times, to avoid exceeding the CAM622's specification for the maximum number of FLASH updates (Table 6). It must not be performed each time the CAM622 is used.

This is not normally an issue for production applications, because a CAM622 IC configured for standalone operation is unlikely to then be subsequently used inside a different product using SPI communication. It is more likely to be encountered during development, when care should be taken.

7.3 Measurement Process

Figure 18 illustrates the process for communicating with the CAM622 over SPI, to configure it and then take repeated measurements.

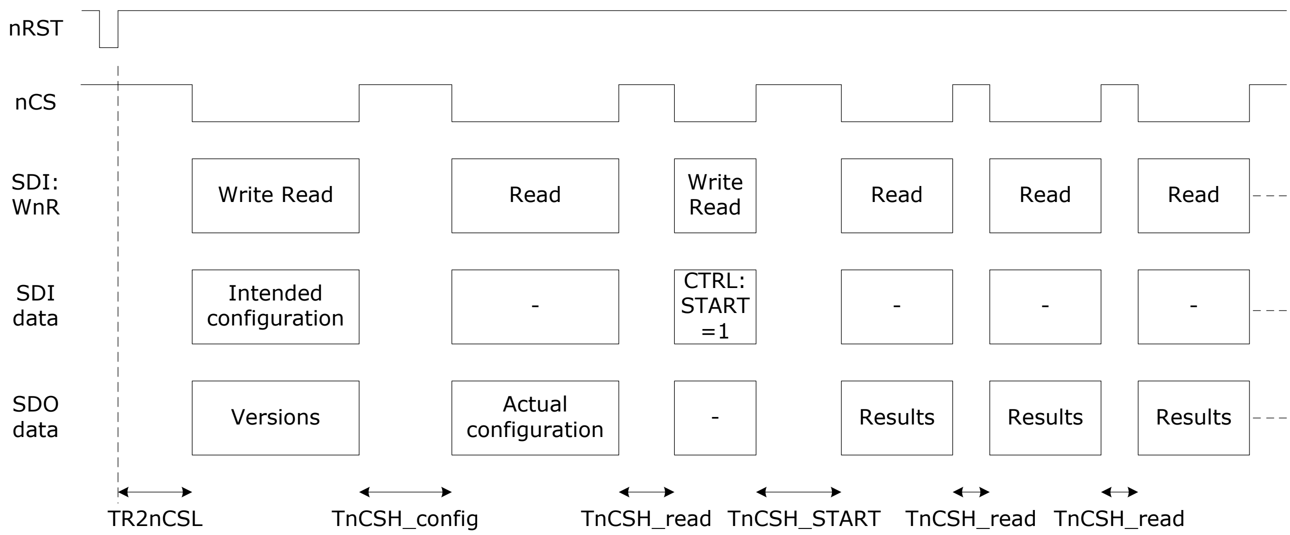


Figure 18 SPI transactions for taking repeated measurements

The host starts the process by toggling nRST low to reset the CAM622. This puts the CAM622 into a known state, including default register values. A delay TR2nCSL(min) is then required before further SPI activity to allow the CAM622's internal self test and validity checking procedure to complete. TR2nCSL is described in section 5.3 and its minimum value is specified in Table 11.

The first SPI transaction is to write the intended register configuration to the CAM622, and read back its version numbers. This SPI transaction accesses all registers up to and including SYSCONFIG, or SICONFIG if Sample Indicators are used (section 7.4). Table 41 lists the registers to be accessed, and how to establish their values.

Table 41 Registers requiring configuration for taking measurements over SPI

Register	Field	Value	Section Ref
CTRL	WnR	1: Write-read transaction.	6.3
	START	0: Do not start measurements (this will be done in a later SPI transaction).	
SECONFIG	DDC	Usually 0. Set to 1 to disable delay compensation.	13.1
	ABSSEL	Usually 15, for 32768 fine measurements between every coarse.	1.2
	SUBTYPE	Set to match sensor's Subtype.	6.4
INTERVAL		It is recommended to use the default value of 0x012C (30µs).	
FLTLVLS	MAXFILTLVL	Usually 255 for maximum filtering.	13
	MINFILTLVL	Usually 255 for maximum filtering.	
HYAD	HYSESET	0: the CAM622 does not apply hysteresis to measurements reported over SPI.	
	ADAPTSENS	Usually 0 for no adaptive filtering, or see reference for how to configure.	13.5
SYSCONFIG	SYSID	Change from default value (0xA) to detect subsequent reset, e.g. 0xB.	6.8
		Write 0s to other SYSCONFIG bits.	
SICONFIG	SIAH	If sample indicators are to be used, select active state of SI signal.	6.9
	SIDO	If sample indicators are to be used, select output type of SI signal.	

This first write-read SPI transaction returns default register values to the host. These include version numbers (Table 14). It is recommended to check returned version numbers against expected values.

A time delay of at least TnCSH_config(min) is required after this first SPI transaction and before any subsequent SPI transaction to give the CAM622 time to implement changes to register settings. This is specified in Table 11.

It is then recommended to perform a read SPI transaction, to verify that the registers and fields listed in Table 41 have been correctly communicated to the CAM622. A delay of at least TnCSH_read(min) is required after this SPI transaction and before the next one. This is specified in Table 11.

Measurements must now be started by writing 1 to the CTRL register's START bit (section 6.3). A delay of at least TnCSH_GO(min) is required after this SPI transaction and before the next one, specified in Table 11.

The host may now perform an SPI transaction accessing the results registers (Table 13) each time new values are required. A delay of at least TnCSH_read(min) is required after this SPI transaction and before the next one. This is specified in Table 11.

Please refer to section 7.5 for how to interpret the values returned in results registers.

7.4 Using the Sample Indicator

The SI Sample Indicator output activates each time new results become available. This is illustrated in Figure 19.

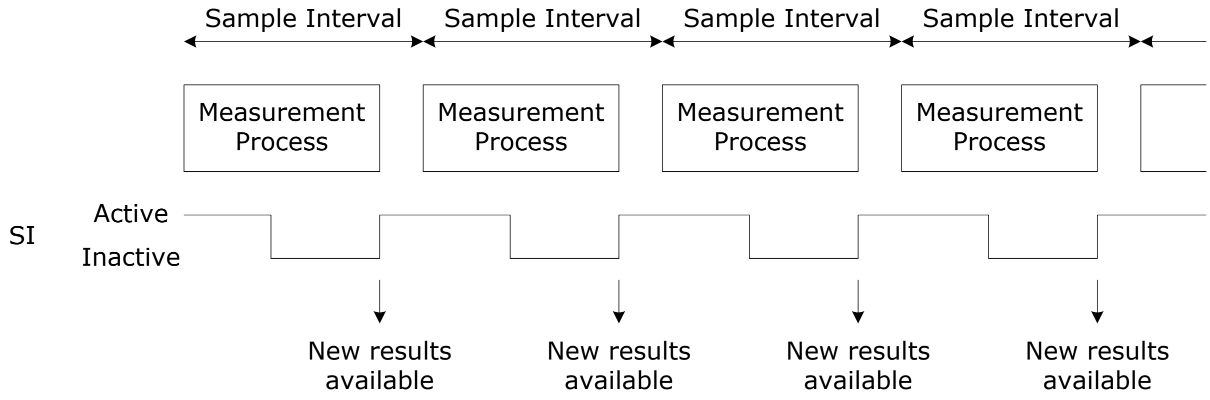


Figure 19 Sample Indicator behaviour

The time between successive measurements is denoted the Sample Interval. It is controlled by the INTERVAL register (section 6.5). When set to the default value, the Sample Interval is minimised at a nominal 30µs.

When each measurement completes, new results become available. If a host system requires a very low and constant latency, it may be designed to initiate an SPI results read immediately after each activation of SI. However for most applications Latch Point Compensation (section 13.6) allows the host to perform SPI transactions asynchronously to SI with no significant performance penalty.

Another application for the Sample Indicator signal is to allow a host to measure the Sample Indicator relative to its own timing reference. This can improve the accuracy of the system's velocity measurement (section 7.5). To take a measurement, the host should measure the time between one transition from inactive to active to the next. For greater accuracy, it can alternatively measure the time between multiple transitions and divide the result by the number of Sample Intervals spanned. The measurement process should be repeated occasionally to ensure that the host's measurement of Sample Interval maintains accuracy in case of drift of the CAM622's on-chip oscillator, especially as temperature changes.

7.5 Interpreting Results

Section 7.3 detailed how a host can read repeated results from the CAM622. This process returns the values of the CTRL register (Table 25) and results registers (Table 13). Each read of results returns values from the most recently completed measurement. This section describes how to interpret these results.

The CTRL:VALID bit indicates whether the result of the most recent measurement was valid (1) or not (0). In order to report a VALID result, the target must be in range of the sensor, both in position and resonant frequency. Once this condition is met, it can take up to 80 measurements for the Motion Filter to become initialised (section 13.1), at which point the CAM622 will report VALID.

When VALID=0, the host should ignore results register values. When VALID=1, values obtained from the results registers are reliable.

The POSH and POSL registers are the high word and low word of a 32-bit position result respectively ("POS32"). In the case of a 360° rotary sensor, this may be converted to position in mechanical degrees using Equation 3.

Equation 3

$$Position = \frac{POS32}{2^{32}} \times 360°$$

POS32 may be interpreted either as an unsigned or as a signed integer. When interpreting POS32 and other CAM622 results as signed integers, they are represented in two's complement notation.

If the host only needs position to 16 bit resolution then the POSL register value can be ignored. In this case the 16-bit POSH value may be converted to mechanical degrees using Equation 4.

Equation 4

$$Position = \frac{POSH}{2^{16}} \times 360°$$

The VELH and VELL registers are the high word and low word of a 32-bit velocity result respectively. These must be interpreted as a signed integer ("VELI32"). This may be converted to mechanical velocity using Equation 5.

Equation 5

$$Velocity/° \; per \; second = \frac{VELI32}{2^{32}} \times 360° \times \frac{1}{Sample \; Interval/seconds}$$

To convert POS32 to a linear dimension for a linear sensor use Equation 6:

Equation 6

$$Position\;/mm = \frac{POS32}{2^{32}} \times Subtype \; \times SinLength/mm$$

Subtype is the sensor's Subtype value. SinLength is its SinLength value (SinLenghA if it is a precision sensor with Subtype greater than 1). For example the 200mm B12 linear sensor has a Subtype of 6 and SinLength of 20.00mm.

To interpret VELI32 as a linear velocity use Equation 7:

Equation 7

$$Velocity/mm \; per \; second = \frac{VELI32}{2^{32}} \times Subtype \; \times SinLength \; in \; mm \times \frac{1}{Sample \; Interval/seconds}$$

Sample Interval is controlled by the CAM622's INTERVAL register (section 6.5). Its default value is nominally 30µs. The nominal value can be used in Equation 5. However in this case the velocity reading will be subject to the CAM622's on-chip oscillator tolerance (see Table 10, "CAM622 Frequency tolerance across parts and Operating Temperature"). In cases where a more accurate reading is required, the actual Sample Interval can be measured by the host relative to its own frequency reference, see section 7.4.

Note that the velocity measurement reported by the CAM622 is NOT equal to the difference between the two most recent position measurements, unless the Motion Filter is disabled (section 13.7). The Motion Filter's estimate of velocity is filtered and delayed more than its position output, as noted in section 13.1. The "reported (over SPI)" trace of Figure 44(c) shows the reported velocity delay in response to a step change in velocity. By way of comparison, the "reported (AB frequency)" plot in that figure illustrates velocity calculated from successive position measurements. The benefit of the velocity measurement reported over SPI is lower noise, providing the Motion Filter is used.

The AMPA register contains the reported Amplitude measurement taken from the sensor's fine coils. This is a measure of sensor signal level, and is an important system health indication. Typical values are shown in a sensor's datasheet. Amplitude varies with gap, and with the presence of nearby metal.

The AMPB register contains a signal level measurement taken from the sensor's coarse coils. The value is not updated each measurement. Instead, it is only updated every coarse measurement. Coarse measurements are scheduled with the ABSSEL bits, see section 6.5. The reported value of AMPB is normally between about 10% and 30% of the AMPA value. AMPB is also important for diagnostic purposes. Basic sensors with Subtype=0 do not include coarse coils and in this case AMPB should be ignored.

The BAMISMATCH register contains the difference between coarse and fine position measurements. This is useful for diagnostic purposes. BAMISMATCH should be interpreted as a signed 16-bit number. Values near zero are an indication of a healthy system. Values whose magnitude exceeds 16000 are an indication of significant mismatch. Like AMPB, BAMISMATCH is only updated every coarse measurement, and it should be ignored for a basic sensor. The FNUM register contains the CAM622's measurement of the target's resonant frequency relative to the Nominal Resonator Centre Frequency (Table 10). An estimate of the resonant frequency, Fres, may be calculated from Equation 8.

Equation 8

$$Fres/kHz = FNUMI16 \times 0.00635 + 833.3$$

The FILTLVL register contains the current filter level setting for the Motion Filter. Please see section 13. Its value can be helpful during development, when setting up Adaptive Filtering (section 13.5). In this case FILTLVL varies between MINFILTLVL and MAXFILTLVL, depending on observed acceleration and noise. It can be useful to check how FILTLVL varies with applied acceleration. It is also important to check that maximum filtering is applied when there is no acceleration and/or motion (FILTLVL remains at MAXFILTLVL). These tests help verify that the sensitivity of the Adaptive Filter (ADAPTSENS) is set appropriately.

The CRC results register is not currently implemented.

8 Using SPI to Configure for Standalone Operation

This section explains how to program a CAM622's configuration into its non-volatile memory over SPI. The CAM622 can then operate autonomously without the need for further SPI connection. Applications include generating quadrature ABN signals (section 9), communication over a BiSS interface (section 10) and generating UVW signals for motor commutation (section 12).

For details of using SPI as the primary communications interface please refer to section 7 instead.

8.1 Programming Connections to CAM622 Circuitry

The CAM622 is configured for standalone operation over its SPI interface. A temporary connection to an SPI host ("programmer") is therefore required for programming. CambridgeIC's MultiComms Adapter may be used as this programmer. Alternatively, a customer may use their own SPI controller device as the programmer.

There are 5 essential programming connections: the 4 SPI lines (nCS, SCK, SDI, SDO) plus 0V. It is strongly recommended to connect nRST too, since this helps the programmer put the CAM622 into a known state. Alternatively, if the programmer also supplies power to the CAM622 circuitry then a power cycle (power on to off to on again) can be used to reset the CAM622 instead of toggling nRST low.

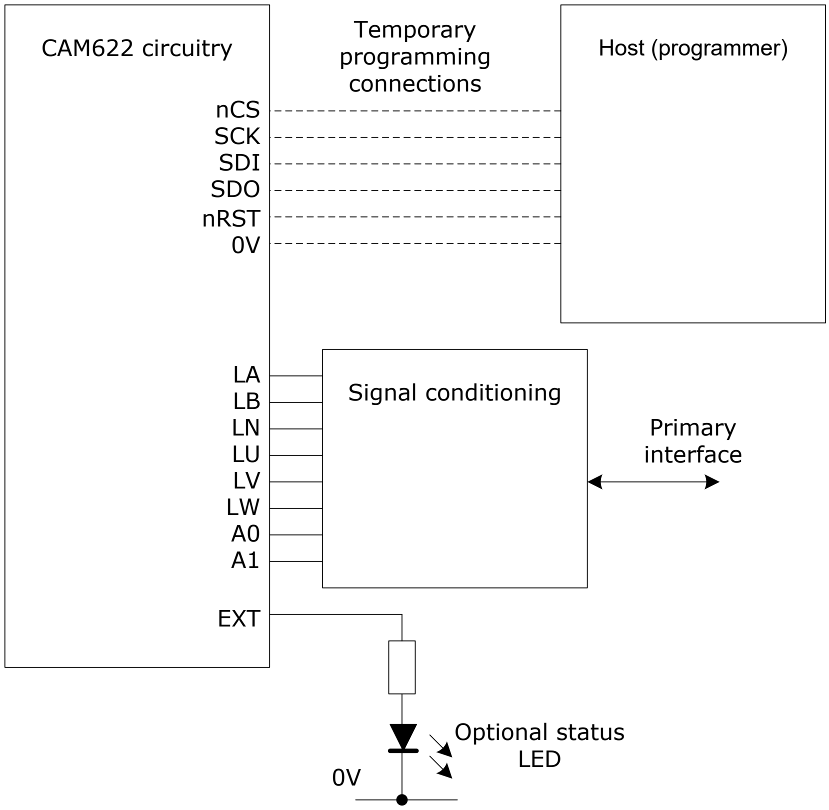


Figure 20 Connections to CAM622 circuitry for standalone operation

Figure 20 illustrates these temporary programming connections, together with a signal conditioning block representing interface circuitry for the chosen communication interface. For circuit options please see 9.5 for ABN, section 10.3 for BiSS and section 12.2 for UVW. The CAM622 can also generate an EXT output signal for controlling an LED.

8.2 Configuration Process

Figure 21 illustrates the process for configuring the CAM622 over SPI.

The host starts the process by toggling nRST low to reset the CAM622. This puts the CAM622 into a known state, including default register values. A delay TR2nCSL(min) is then required before further SPI activity to allow the CAM622's internal self test and validity checking procedure to complete. TR2nCSL is described in section 5.3 and its minimum value is specified in Table 11.

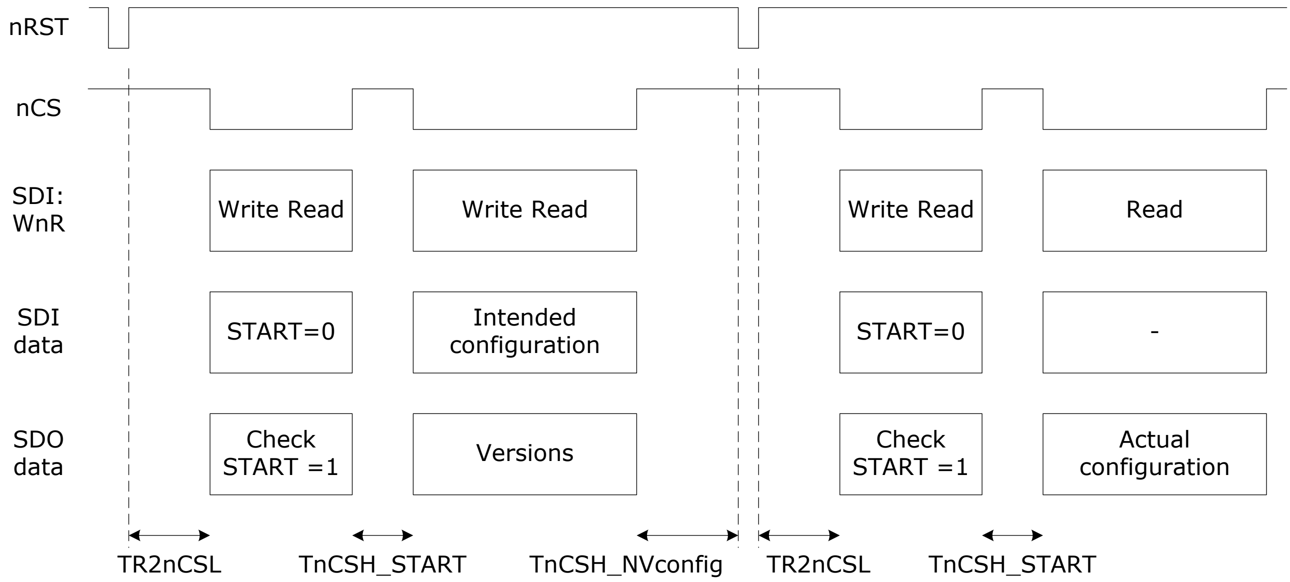


Figure 21 SPI transactions for standalone configuration

The first SPI transaction is a single word write read to clear the START bit in the CONFIG register. This is required in cases where the CAM622 has already been configured for standalone operation. In this case the CAM622 will start taking measurements following a reset, because START is always set to 1 for standalone operation. Clearing START allows the host to access all configuration registers in a subsequent SPI transaction, at least TnCSH_START later. This step is not necessary if the CAM622's START bit is already 0, for example if it has never been programmed with non-volatile defaults before or if factory defaults were restored immediately beforehand (section 8.5).

The next SPI transaction writes the intended register configuration to the CAM622's non-volatile memory, and reads back its version numbers. Table 42 summarises the settings that apply to all interfaces. If ABN encoder outputs are required please also refer to Table 43. If BiSS interface operation is required please also refer to Table 45. If a SENT interface is required, please refer to Table 56. If motor commutation UVW outputs are required please also refer to Table 63. The SPI transaction must access all of the registers in order, up to and including the last register associated with the chosen interface(s).

This write-read SPI transaction returns default register values to the host. These include version numbers (Table 14). It is recommended to check returned version numbers against expected values.

A time TnCSH_NVconfig(min) is required after this SPI transaction in order for the CAM622 to update register values and write them to non-volatile memory. Please refer to Table 11 for its value.

The CAM622 performs an internal reset once this process is complete. The CAM622's default register values should now include the newly configured register values. It then begins standalone operation once out of reset.

It is recommended to perform a verification step, to check that registers have been updated correctly and in a non-volatile way. To do this, the programmer should first toggle nRST low once the TnCSH_NVconfig period has elapsed. The ensuing reset ensures that register values subsequently read out of the CAM622 are truly non-volatile. A time TR2nCSL(min) is then required before further SPI activity to allow the CAM622's internal self test and validity checking procedure to complete.

Next, a write-read SPI transaction accessing only CTRL should be used to set CTRL:START=0. This allows the host to read out the values of registers beyond address 0x0E in a subsequent SPI transaction. This transaction returns the value of CTRL:START stores in the CAM622's non-volatile memory, and the host should check this is 1.

Finally, now that START=0, the host can use a read transaction to read the CAM622's configuration and check register values against expected values (Table 42).

To resume autonomous measurements and outputs, the host can either reset the CAM622 or use a write-read SPI transaction to set CTRL:START=1. It is recommended that these outputs (e.g. ABN encoder signals) be tested, by connecting appropriate measurement equipment to the relevant outputs.

Table 42 Registers requiring configuration for autonomous operation

Register	Field	Value	Section Ref
CTRL	WnR	1: Write-read transaction.	6.3
	START	1: Start measurements.	
POSH		0x0C1C: required for the non-volatile SAVE function to take effect.	6.2
SECONFIG	DDC	Usually 0. Set to 1 to disable delay compensation.	13.1
SECONFIG	ABSSEL	Usually 15, for 32768 fine measurements between every coarse.	1.2
SECONFIG	SUBTYPE	Set to match sensor's Subtype.	6.4
INTERVAL		The default value of 0x012C MUST be used (30µs sample interval)	6.5
FLTLVLS	MAXFILTLVL	Usually 255 for maximum filtering, or select other value.	13
	MINFILTLVL	Usually 255 for maximum filtering, or select other value.	
HYAD	HYSESET	Set to 0 for no hysteresis, or select amount of hysteresis applied to ABN and UVW signals.	9.7
	ADAPTSENS	Usually 0 for no adaptive filtering.	13.5
SYSCONFIG	SYSID	Change from default value (0xA) to indicate programming, e.g. 0xC.	6.8
	SAVE	1: save these register values to non-volatile memory.	
	A0, A1	Sets digital states of A0, A1 pins e.g. to configure external circuitry.	
	EXTMODE	Set to LED control if LED connected for visual indication of status.	
		Write 0s to other SYSCONFIG bits.	
SICONFIG	EXTAH	Set to active state for LED on if used.	6.9
	LEDTHRESHOLD	Set to maximum value of AmplitudeA for flashing LED.	

8.3 "Teach" Step

It may be desirable to establish the value of some parameters by experiment, during a "teach" step at manufacture or installation. For example the ideal value of NPOS may depend on the orientation of a target on a shaft, which may depend on how it was installed. In this case the programmer can first take position measurements over SPI and read results, as detailed in section 7. It can then use those results to calculate appropriate values for standalone operation.

For example, to teach the position where the N signal activates, move the target to the desired position for ABN count zero (Figure 24). Take readings from POSH, ideally averaging a number of readings (modulo 65536) to eliminate the effect of noise. Use the resulting value for NPOS when configuring for standalone operation.

8.4 Testing Configurations Without Updating Non-Volatile Memory

In some situations it may be helpful to test the effect of different register settings iteratively. Multiple updates to non-volatile memory should be avoided, because there is a maximum number for any CAM622 part. This number is large, see section 2.5. However it is good practice to avoid multiple updates to non-volatile memory where possible, to make it easier to verify that the programming process never exceeds the limit.

To avoid updating non-volatile memory each time the CAM622's standalone configuration is updated, use the settings listed in Table 42 except with SYSCONFIG:SAVE set to 0 and POSH set to 0x0000. Then test the system without performing a reset. Once an optimum configuration is achieved the register settings can be made non-volatile using the process detailed in section 0.

8.5 Restoring Factory Defaults

The configuration process detailed in section 0 leaves a CAM622 in a custom state which survives a reset. There is a mechanism to return a CAM622 to factory defaults, so that register values return to their default values following a reset. This can be used to prepare a CAM622 for communication using SPI as a primary interface, see section 7.2.

To restore a CAM622's registers to factory defaults, reset the CAM622 and then perform a write-read SPI transaction with POSH set to 0x0C1C and the SYSCONFIG:FACTORY bit set to 1 (section 6.8). This may be followed by another reset and then a read SPI transaction to verify the change. This verification step can include checking that CTRL:SYSID has returned to its default value of 0xA, providing a different SYSID value was programmed as part of any custom configuration as recommended in Table 42. The process is the same as the one illustrated in Figure 21, including timings.

9 Encoder Operation

The CAM622 may be configured to generate digital ABN outputs like an optical encoder. A and B signals are a quadrature count, and N is an index pulse that asserts once per revolution.

The CAM622 is configured for autonomous operation over its SPI interface. Please refer to section 8. This includes Table 42 which lists the registers requiring configuration for autonomous operation.

9.1 Settings Relevant to Encoder Signals

This section describes available settings for the ABN output. These are typically written to the CAM622's non-volatile memory in a configuration step at manufacture, see section 8. Once programmed in this way the CAM622 operates autonomously with the chosen settings.

Table 43 lists the available settings associated with the ABN encoder output. The following subsections detail their effects.

Table 43 Encoder settings

Register	Field	Bits	Function	Section Ref
ABCONFIG	ABEN	1	Set this bit to 1 to enable encoder operation.	6.11
	NSTART	1	Selects whether or not ABN counts to correct absolute position when first VALID. If 1, N pulse activates then AB counts to current absolute position. If 0, AB counts from current position, like a conventional optical encoder.	9.8
	ABDIR	1	Reverses AB count direction when set.	9.6
	ANEG	1	Polarity of logic output LA relative to A. If ANEG=1 LA is an inverted version of A.	9.4
	BNEG	1	Polarity of logic output LB relative to B. If BNEG=1 LB is an inverted version of B.	
MAXABFREQ		8	Controls the maximum number of AB edges per measurement, and hence maximum ABN frequency.	9.8
NPOS		16	Controls the position of the N pulse, which is adjustable across 360° of mechanical rotation.	9.6
ABCYC		14	Controls the number of AB cycles around 360°. Set to ABCYC=0 for 2^{14} =16384 cycles.	9.6
HYAD	HYSTSET	8	Controls the amount of added position hysteresis, used to control ABN edge flicker if needed.	9.7
INDEX10		16	Set to 0x1817 so INFO0 and 1 contain ABCOUNTH and ABCOUNTL	6.10

The ABN encoder outputs are also affected by the Motion Filter described in section 13, and its settings are listed in Table 64. These settings affect the dynamic behaviour of the position values fed into the Encoder Edge Control Loop. Filtering can help reduce noise. This in turn can help control ABN edge flicker if needed for achieving high resolutions.

The Motion Filter's DDC setting ("Disable Delay Compensation") controls whether the Motion Filter compensates for measurement and filter delays. It is also used to control delay compensation applied to the Encoder Edge Control Loop. When set to the default value of 0, the CAM622 compensates for delays both in the Motion Filter itself and in the Encoder Edge Control Loop. When set to 1, this compensation is disabled, resulting in the delays specified in Table 65.

9.2 Effect of ABCYC

ABCYC controls the number of AB cycles per 360° target revolution according to Equation 9:

Equation 9

$$AB\ Cycles\ Per\ Rev = \ 16384\ IF\ ABCYC = 0$$
$$AB\ Cycles\ Per\ Rev = \ ABCYC\ IF\ ABCYC > 0$$

There are 4 AB edges per AB Cycle, so the number of AB edges per 360° target revolution is given by Equation 10:

Equation 10

$$AB\ Edges\ Per\ Rev = 4\ \times AB\ Cycles\ Per\ Rev$$

Note that an encoder's Interface Resolution is usually expressed as the number of distinguishable states, hence Equation 11:

Equation 11

$$Interface\ Resolution = \ log_2\,(AB\ Edges\ Per\ Rev)$$

9.3 True Edge Timing

The Interface Processor includes an Encoder Edge Control Loop as illustrated in Figure 5. Its purpose is to synthesise ABN signals whose edges match as closely as possible those generated by an ideal optical encoder.

The Encoder Edge Control Loop updates every 30µs on a new position measurement. This is set by the INTERVAL register, and the default value must be used for AB signal generation, see section 6.5. The control loop aims to maintain a match between actual position and ABN count value, by controlling the frequency and timing of ABN edges.

When there is more than one ABN count change per measurement, the count rate is constant between measurements. An ideal encoder's count rate changes continuously, so there can be a small apparent ABN count error during periods of extremely high acceleration. This error is tiny for practical purposes, due to the very small time between measurements.

9.4 LA and LB Signal Polarity Control

LA and LB may optionally be inverted using the ANEG and BNEG bits. This function is illustrated schematically in Figure 22.

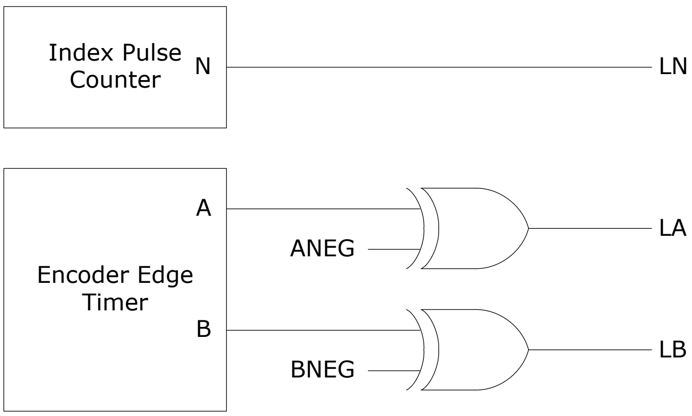


Figure 22 Effect of ANEG and BNEG

LN always equals the state of N.

9.5 Interfacing to LA, LB and LN

The CAM622 chip's encoder outputs (LA, LB, LN) are 3.3V nominal logic level and have a relatively low output voltage and current (section 2.4). These may be sufficient where they can be directly connected to a 3.3V powered host device on the same PCB or with short and direct connections. However some applications will require signal transmission over longer cable lengths, with greater voltage and peak current. This requires line drivers to be added, usually driven from a different power supply voltage.

Encoder signals are often transmitted differentially to the RS-422 standard, requiring differential outputs. This can be done with a Line Driver IC with differential outputs, as illustrated in Figure 23. These are widely available, for example the iC-DL from iC-Haus.

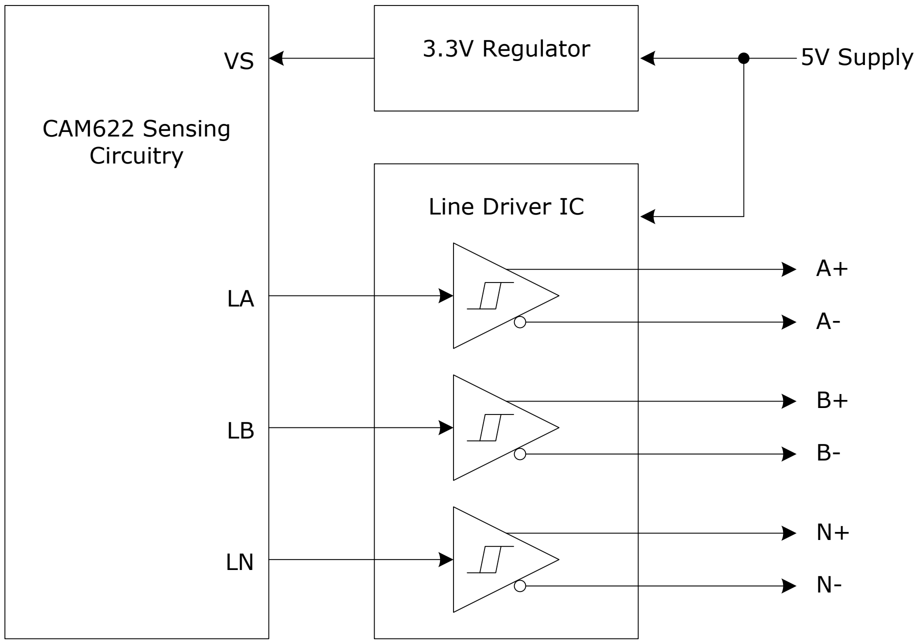


Figure 23 Generating differential ABN outputs using Line Drivers

Encoders are often operated from a 5V power supply. The CAM622 circuitry is supplied from 3.3V, so a regulator is also usually required.

Figure 33 illustrates an alternative scheme using line transceivers. This allows configuration over the ABN signals, and also supports BiSS communication.

9.6 Effect of ABDIR, NPOS and ABCYC

Figure 24 illustrates how A, B and N signals vary with position when enabled (ABEN=1). The A and B signals are a quadrature count, and N is an index pulse that asserts once per revolution.

Figure 24(a) illustrates the signals when ABDIR=0. In this case the A and B signals count in a positive direction when the target's position moves in the positive direction relative to the sensor. A positive count direction is defined as A leading B. Please refer to the Type B Sensor Reference Manual for a definition of the coordinate system used, including Actual Angle and its direction. Figure 24(b) illustrates the signals when ABDIR=1. In this case the A and B signals count in a negative direction when the target's position moves in the positive direction.

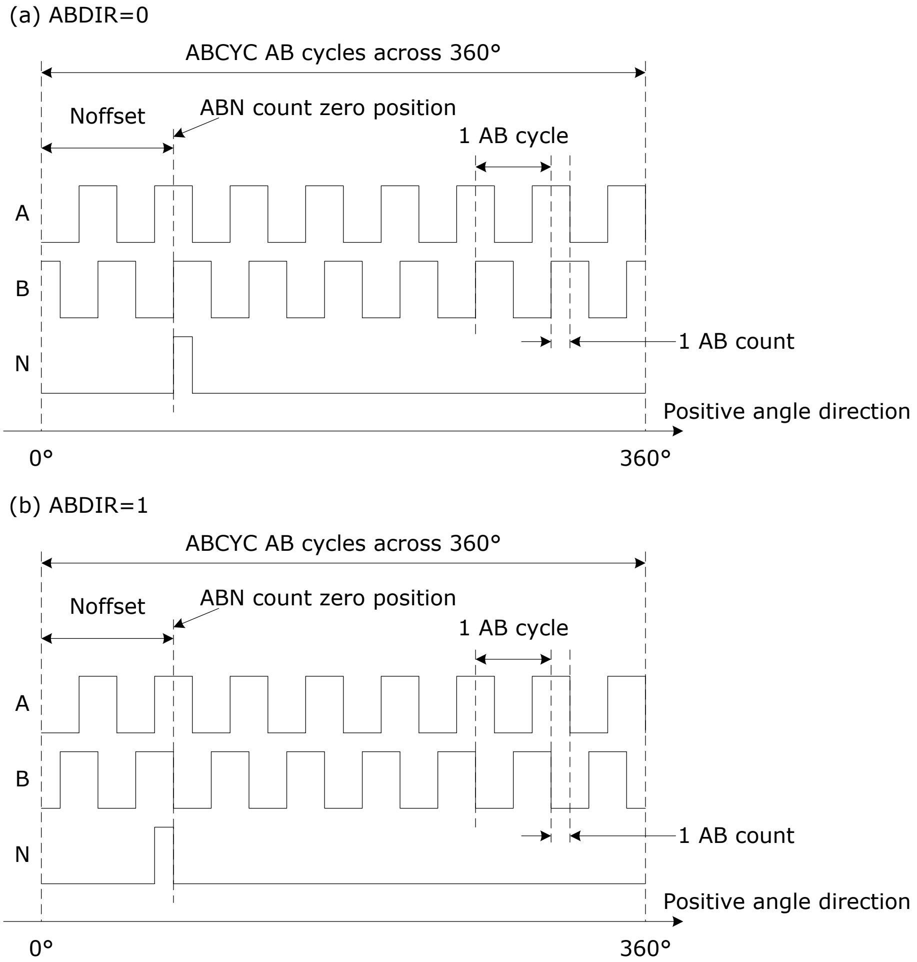


Figure 24 ABN signals as a function of position

The ABCYC parameter controls the number of AB cycles around 360°. This is marked in Figure 24, which illustrates the case when ABCYC=8. Set ABCYC to the number of cycles required, or to 0 if 16384 cycles are required.

An AB Count is the smallest angle change that can be directly measured from ABN signals, and is marked in Figure 24.

The ABN count zero position is variable across 360° and is controlled by the NPOS setting. NPOS is a 16-bit integer value. The relationship between NPOS and the physical angle Noffset shown in Figure 24 is given by Equation 12.

Equation 12

$$Noffset = \frac{NPOS}{65536} \times 360°$$

Note that the N signal asserts at the phase of the AB cycle when A and B are both 1. Some hosts may require the opposite, with N asserting when A and B are both 0. In this case, use the ANEG and BNEG bits (section 6.11) to invert LA and LB relative to A and B (section 9.5).

9.7 Applying Hysteresis with HYSTSET

Hysteresis may be deliberately added to ABN encoder signals in order to reduce or eliminate edge jitter. A conventional optical encoder's photodetectors include hysteresis for the same reason, although in this case it is not controllable.

The effect of hysteresis is to introduce a position lag between actual angle and encoder count. This is illustrated in Figure 25. The AB count value trails actual position for movement in both directions. This means there is a hysteresis band between the transfer function in each direction. Its size is denoted Position Hysteresis when measured in angle units (°) and AB Count Hysteresis when measured in AB counts.

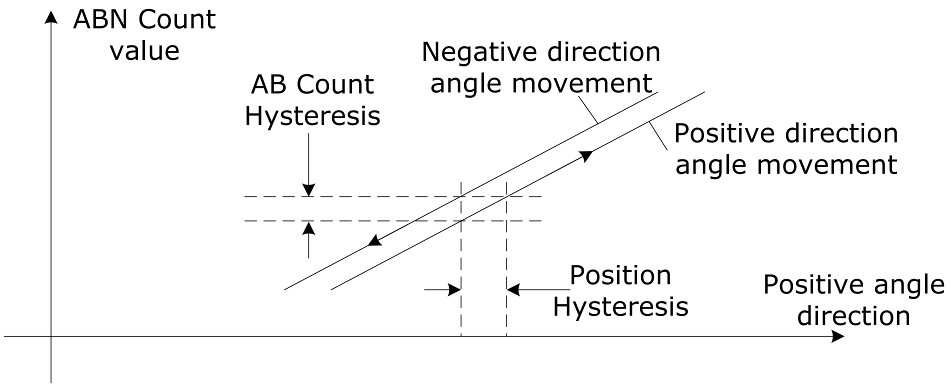


Figure 25 Definition of Position Hysteresis and AB Count Hysteresis

Figure 26 illustrates the effect of hysteresis on A and B encoder signals. The solid waveforms are for a positive count direction and the dashed lines are for a negative count direction. Strictly, these are nominal waveforms in the absence of noise. Noise will tend to affect the precise location of each edge. The reason to add hysteresis is to help reduce or eliminate edge jitter, especially around zero speed including when the angle is reversing. To eliminate it completely, the peak to peak position noise (after filtering, if used) should be less than the Position Hysteresis.

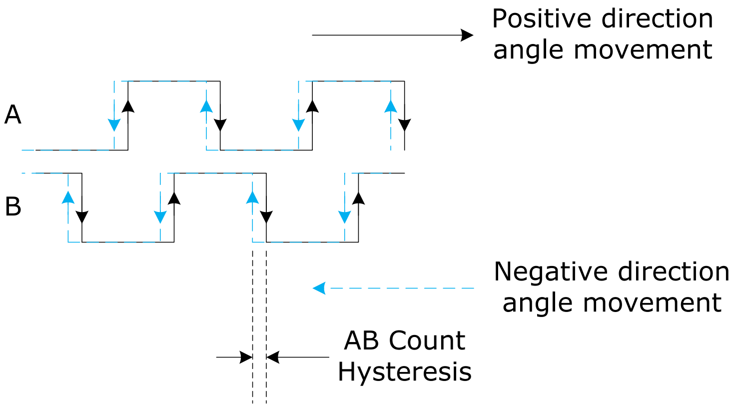


Figure 26 Effect of hysteresis on encoder signals

To eliminate edge jitter, first establish worst case peak to peak position noise, and hence a Position Hysteresis value. This may be done by direct measurement of a physical system, including appropriate Motion Filter Settings which also reduce noise. It may also be done by analysis, by taking Noise Free Resolution values from a sensor's datasheet and adding the filter's resolution improvement from Figure 43.

Next, read off the value of HYSTSET corresponding to this Position Hysteresis value from Figure 27.

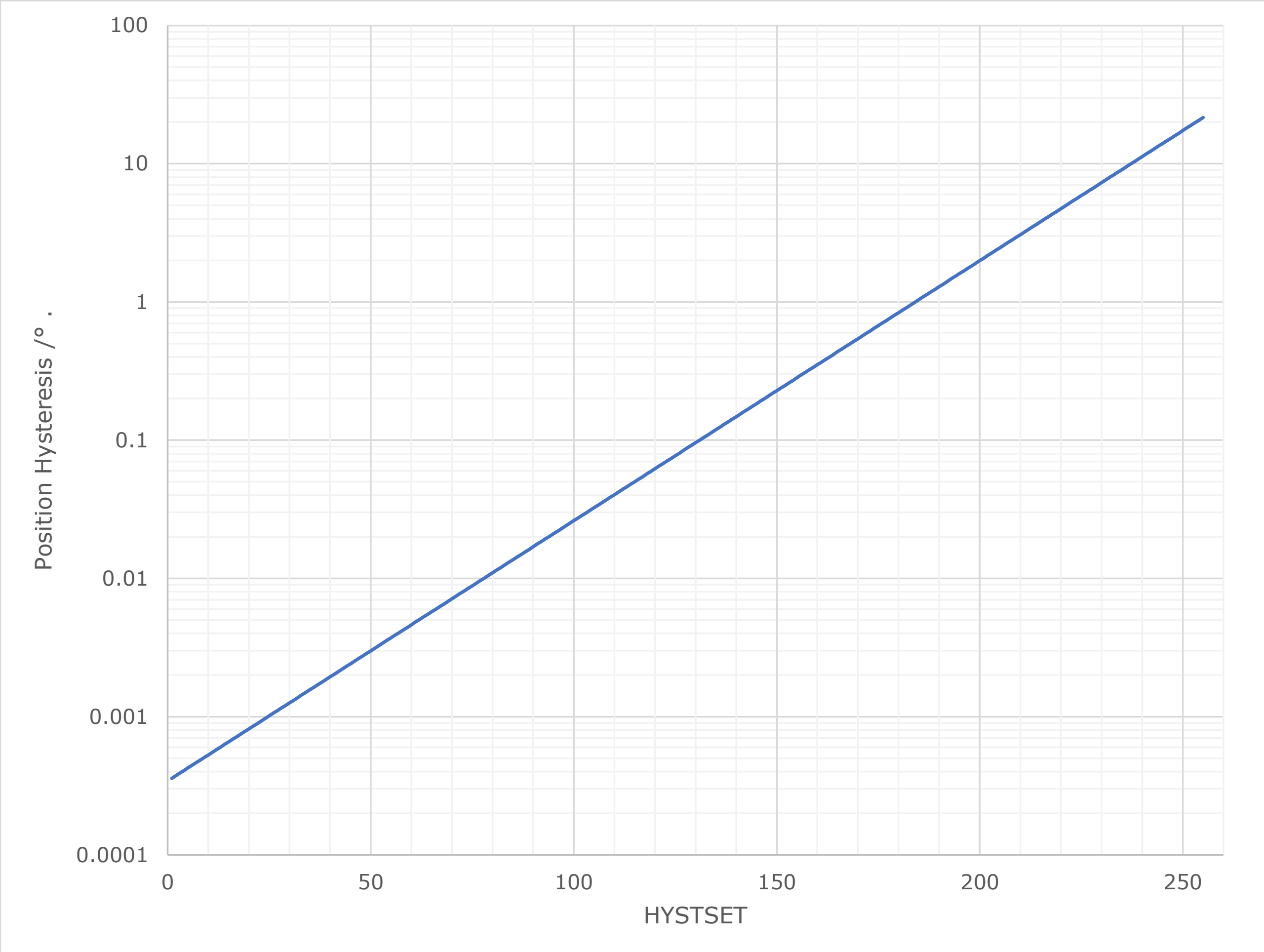


Figure 27 Position Hysteresis as a function of HYSTSET

Position Hysteresis may also be calculated from AB Count Hysteresis, in cases where it is easier to define hysteresis in AB Counts:

Equation 13

$$Position\ Hysteresis\ /° = \frac{AB\ Count\ Hysteresis}{4\ \times cycles\ per\ revolution} \times 360°$$

AB Count Hysteresis is usually less than 1 count, although this is not essential.

Note that it is not always important to add hysteresis. The host device's counter may function correctly when there is AB edge jitter. Edge jitter must usually be avoided in optical encoders because the frequency of the jitter may exceed the host device's ability to count it correctly. However the CAM622 enforces a minimum time period for a count change when it reverses direction. This means that the N pulse is never shorter than 80ns, which helps ensure that the host's count remains correct.

9.8 Effect of NSTART and MAXABFREQ

The CAM622's ABN output normally emulates the output of a conventional optical encoder. However unlike an optical encoder, the CAM622 "knows" absolute position when its measurements are VALID. It may be configured to count to the current position from the reference position (N active) when first VALID. This has the effect of initialising a host's ABN counter to the correct absolute position following power on. To configure this mode of operation, the NSTART bit should be set. Figure 28 illustrates CAM622 behaviour in this case.

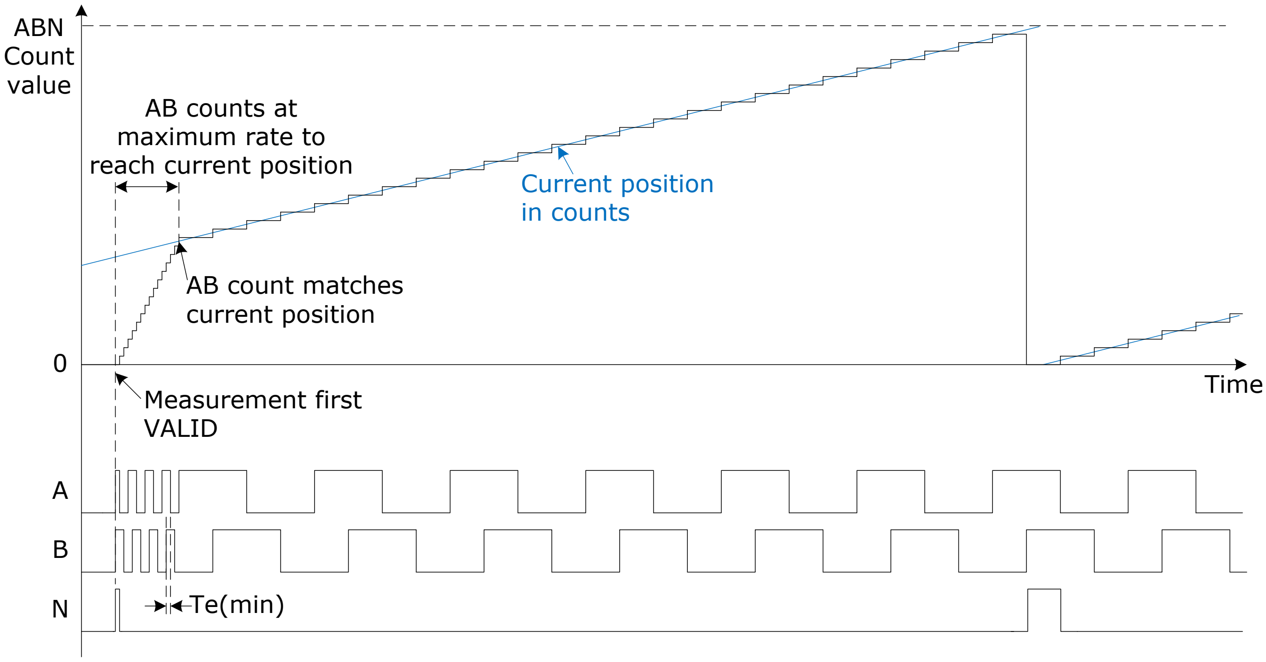


Figure 28 ABN behaviour following first VALID with NSTART=1

When NSTART=1 and first VALID, the CAM622 asserts N. The A and B signals then toggle at a maximum rate until the indicated count value reaches the actual position in counts. The A and B signals then continue to count to match actual position. A host system counter is therefore initialised with the correct absolute position, without the target having to rotate past the reference position.

If the CAM622's measured position subsequently becomes invalid the A and B signals will stop counting. When VALID again, they will count from the AB count value immediately preceding invalid. If the target moved while invalid, the AB count will transition at maximum rate as before once VALID again, to ensure that the host's AB count becomes correct as soon as possible afterwards.

The CAM622 selects the initial count direction to minimise the number of AB transitions required to reach the current position from the reference position when first VALID. Note that this direction may be the opposite direction to any actual target rotation.

Te is the Edge Interval, the time between A and B edges. The AB edge rate is defined as its reciprocal: 1/Te. As in Equation 14:

Equation 14

$$AB\ Edge\ Rate = {}^{1}/_{Te}$$

If a host system can tolerate a certain Maximum AB Edge Rate, the value of MAXABFREQ required is given by:

Equation 15

$$MAXABFREQ = \frac{Maximum\ AB\ Edge\ Rate}{34.3kHz}$$

The maximum allowed value of MAXABFREQ is 141, corresponding to a Maximum AB Edge Rate of 4.8MHz. If a host system can tolerate an AB Edge Rate of more than 4.8MHz then set MAXABFREQ to 141.

When NSTART is set to 0, the CAM622 counts from the current position when first VALID. This matches the behaviour of an optical encoder when it is first switched on.

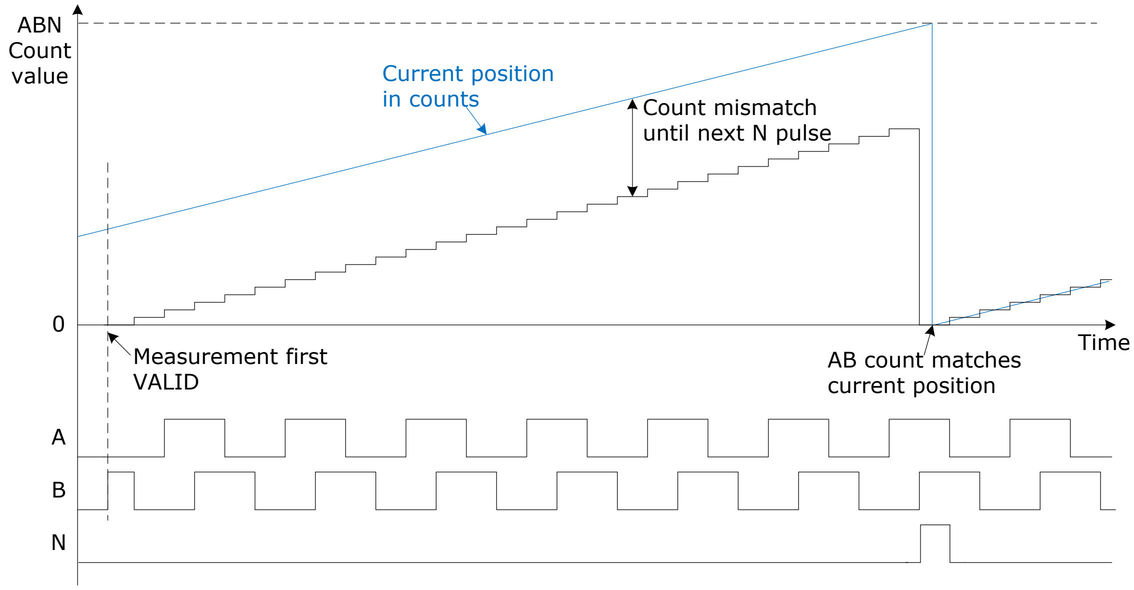


Figure 29 ABN behaviour following first VALID with NSTART=0

When NSTART=0, the host's AB count value does not match the current position until the target has rotated past the reference position. This matches the behaviour of a conventional optical encoder and most other encoder products.

MAXABFREQ control sets the maximum AB rate for all AB edges at all times, including NSTART=0 and NSTART=1. It does not only apply following first VALID when NSTART=1. For the ABN interface to operate correctly, the AB edge rate due to normal target rotation must always be less than the value given by Equation 16:

Equation 16

$$Maximum\ Rotation\ Rate\ /rpm = 60\ \times \frac{MAXABFREQ\ \times 32300Hz}{4\ \times AB\ Cycles\ Per\ Rev}$$

This yields a maximum rotation rate of 4170rpm with the maximum MAXABFREQ=141 and 16384 cycles per rev.

Note that the 32.3kHz figure given in Equation 16 is less than the 34.3kHz figure given in Equation 15. The difference is due to the CAM622 Frequency tolerance specified in Table 10. Equation 15 uses a nominal figure of 33.3kHz plus an allowance for tolerance, to yield a MAXABFREQ to guarantee that the Maximum AB Edge Rate is less than or equal to the required value. Equation 16 uses a nominal figure of 33.3kHz minus an allowance for tolerance, so that the Maximum Rotation rate that is calculated is always less than the maximum that can be generated, across the CAM622 frequency tolerance range.

9.9 N Pulse Timing

The CAM622's N pulse is delayed slightly relative to the corresponding A and B edges, as illustrated in Figure 30.

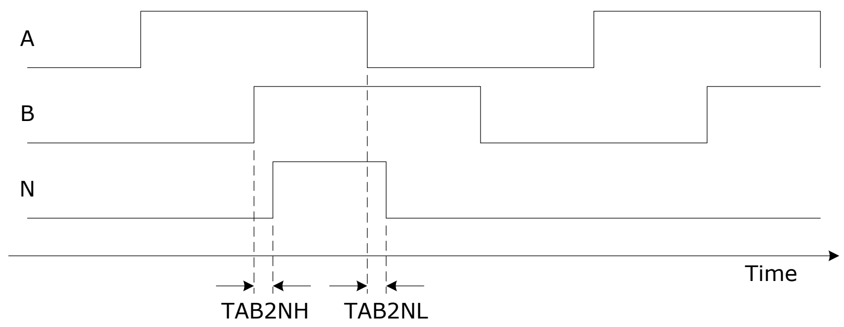


Figure 30 N Edge Delay Parameters

Figure 30 illustrates A, B and N signals for clarity. However the timing parameters actually apply to the LA, LB and LN signals derived from ABN as illustrated in section 9.5 TAB2NH is the delay time between A and B both becoming high and N going high. TAB2NH is the delay time between one of A or B going low again and N going low. Table 44 lists maximum values for these parameters.

Table 44 N Edge Delay Specifications

Parameter	Max value
TAB2NH	50ns
TAB2NL	50ns

9.10 ABN Accuracy

There are two main aspects to ABN signal accuracy:

- ABN Spatial Accuracy
- ABN Edge Jitter

ABN Spatial Accuracy concerns the accuracy with which ABN edges match the physical positions that they represent. It is largely a function of the sensor's linearity error. Please refer to the sensor datasheet for more information. When operating at high rotation speeds with a high filter level, the motion filter may act to filter linearity error, yielding somewhat improved ABN Spatial Accuracy.

The Edge Interval Te is already defined in Figure 28, and is shown again in Figure 31 for clarity.

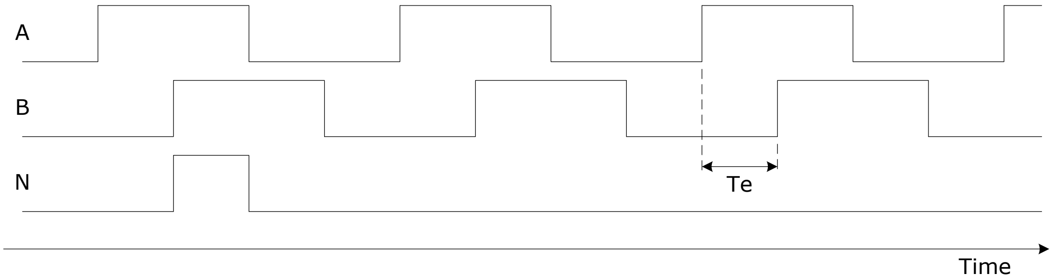


Figure 31 Edge Interval

Peak Te Jitter is defined as the magnitude of the difference between minimum and maximum values and the average, as in Equation 17:

Equation 17

$$Peak\ Te\ Jitter = Te(max) - Te(average) = Te(average) - Te(min)$$

In normal operation with a healthy signal amplitude (AmplitudeA≥4000), and velocity ≥1000rpm, the peak jitter in the Edge Interval Te may be approximated by Equation 18:

Equation 18

$$Peak\ Te\ Jitter = 15ns + Te \times 0.01$$

If a host system were to measure the time between each AB edge to determine velocity then the resulting velocity jitter as a fraction of velocity may be estimated from Equation 19:

Equation 19

$$Peak\ Instantaneous\ Velocity\ Jitter = \frac{Peak\ Te\ Jitter}{Te\ average} = 1\% + \frac{15ns}{Te} \times 100\%$$

For example, if the target is rotating at 1000rpm with 65536 edges per revolution then Te=900ns. In this case the Peak Te jitter will be approximately 24ns, which is 2.7%.

When velocity is less than 1000rpm or AmplitudeA is low, additional velocity jitter due to position noise may become significant, especially when the number of AB Cycles Per Rev is large.

Note that most well designed host systems will not use a single measurement like this to determine velocity. They will typically measure the time between multiple edges, or the number of edges in a given time. In this case the apparent velocity jitter determined by the host will be lower than given in Equation 19.

9.11 Reading Position in AB Counts over SPI

The POSH and POSL registers together return a 32-bit position value with full scale representing 360° (Equation 3). It may be helpful to instead read position in AB counts, for example while performing iterative testing as described in section 8.4. This is possible when the CAM622's ABN encoder outputs are enabled (ABEN=1). Values appear in the ABCOUNTH and ABCOUNTL registers (Table 18). Since these registers have addresses beyond 0x0E, a host SPI device can not read them directly when START=1. Instead, they may be accessed through the Information Registers. When configured with the INDEX10 value shown in Table 43, ABCOUNTH will appear at address 0x0B (INFO0) and ABCOUNTL at 0x0C (INFO1).

ABCOUNTH contains the whole number of AB counts and ABCOUNTL the fractional part. The relationship between mechanical position and counts is given by Equation 20:

Equation 20

$$Position = \left[\frac{NPOS}{2^{16}} + \frac{2^{16} \times ABCOUNTH + \ ABCOUNTL}{2^{16} \ \times \ (4 \ \times ABCYC)}\right] \times 360°$$

NPOS specifies the ABN count zero position offset and ABCYC the number of AB cycles per revolution, see section 9.6. Note that (4 x ABCYC) is the number of AB states or edges per revolution.

To read ABCOUNTH and ABCOUNTL registers, perform an SPI read transaction up to and including INFO1. A delay of at least TnCSH_ABNread(min) is required after this SPI transaction and before the next one. This is specified in Table 11.

Please note that the checksum CRC register is not updated when ABEN=1.

10 BiSS Operation

10.1 BiSS Introduction

References to BiSS in this document are to BiSS C.

BiSS is a standardised method of communication overseen by the BiSS Association, helping to ensure interoperability of host devices and sensors. It is a bidirectional serial digital interface that can interface between a host and sensor device. It uses differential RS422 style signal transmission. This makes it a good choice for industrial applications, especially when the host and sensor devices are far apart. By contrast, SPI is usually only suitable when the host and sensor devices are physically close, typically on the same PCB.

The CAM622 supports BiSS point to point using an external iC-MCB bridge IC available from iC-Haus. This manages the timing for the BiSS frames and management of the lower levels of data, along with optional line drivers for the incoming MA and outgoing SLO signals. The CAM622 configures the iC-MCB to interrupt when data is required, both for single-frame position data and multi-frame diagnostic data and configuration.

Together, the CAM622 and iC-MCB transmit position measurements back to the BiSS host at high speed ("Process Data" in BiSS terminology), with a standard format ("BiSS Profile"). They also support bidirectional data transfer at a slower rate ("Control Communication"). This may optionally be used to transmit data from the host to configure the CAM622, and to transmit configuration and diagnostic data from the CAM622 back to the host.

This section describes how to use the CAM622 to interface to a host device over BiSS. Please also refer to the BiSS Protocol Description published by the BiSS Association, and the datasheet for the iC-MCB IC published by iC-Haus.

10.2 Settings Relevant to BiSS

This section describes available settings for interfacing over BiSS. These are typically written to the CAM622's non-volatile memory in a configuration step at manufacture, see section 8. Once programmed in this way the CAM622 operates autonomously with the chosen settings. Table 45 summarises settings relevant to BiSS.

Table 45 Settings relevant to BiSS

Register	Field	Bits	Function	Ref
ABCONFIG	ABEN	1	ABN Interface enable bit	Table 46, 6.8, 6.13
SYS CONFIG	A0 (2)	1	Controls state of A0 or defines active state if A0CTRL=1	
	A1 (2)	1	Controls state of A1 or defines active state if A1CTRL=1	
	A0CTRL (1)	1	When 1 gives CAM622 control over A0	
	A1CTRL (1)	1	When 1 gives CAM622 control over A1	
BISS CONFIG	BISSEN (1)	1	Set this bit to 1 to enable BiSS	
	ESE (1)	1	Controls the iC-MCB's Enable Single Ended bit	
	CB_SLO (1)	1	If set and in BiSS mode the iC-MCB's IO5 mirrors SLO	
	EDS	1	Must be set to 0	6.13
	BISSFORMAT	4	Set to 0 to format the BiSS Frame according to section 10.6	
BISSKEY0 BISSKEY1	BISSKEYCODE	8	Change to 0xC0 to update BISSKEY	6.14
	BISSKEY	24	Locks access to settings over BiSS, or set to 0 to allow	
BISSMID		16	Controls the Manufacturer's ID reported over BiSS Allocated by the BiSS Association	
BISSDID2 BISSDID1 BISSDID0	BISSDID	48	Controls the Device ID reported over BiSS	
BISSSNUM1 BISSSNUM0	BISSSNUM	32	Controls the Serial Number reported over BiSS	

Note (1): This setting can not be modified over BiSS (section 6.8). This is to prevent a host from changing settings over BiSS that might render BiSS unusable, potentially rendering the product unusable.

Note (2): The A0 setting can only be modified over BiSS if A0CTRL=0, and The A1 setting can only be modified over BiSS if A1CTRL=0. This is also to prevent a host from changing settings over BiSS that might render BiSS unusable.

DEVICEID represents the type of the end device and is managed by the device manufacturer. Along with the MANUFID value, the DEVICEID should be able to be used to identify a particular configuration of device (e.g. data length or timings) where such configuration is provided externally (such as via an XML file).

Position values reported over BiSS are also affected by the Motion Filter described in section 13, and its settings are listed in Table 64. Filtering can help reduce noise at the expense of dynamic behaviour.

The Motion Filter's DDC setting ("Disable Delay Compensation") controls whether the Motion Filter compensates for measurement and filter delays. When DDC=0 the Motion Filter applies Latch Point Compensation to BiSS transactions, so that the position values reported over BiSS correspond to position at the BiSS Latch Point (section 13.6). When DDC=1 the system reports the most recent measured position. Please see Table 65 for related timings.

10.3 Schematics for BiSS Support

Section 3 details the circuitry required for basic sensing operation. This section describes the additional circuitry required to support BiSS and optionally both BiSS and ABN.

Figure 32 shows how to connect the CAM622 to an external iC-MCB bridge IC from iC-Haus, in the case that only BiSS support is required.

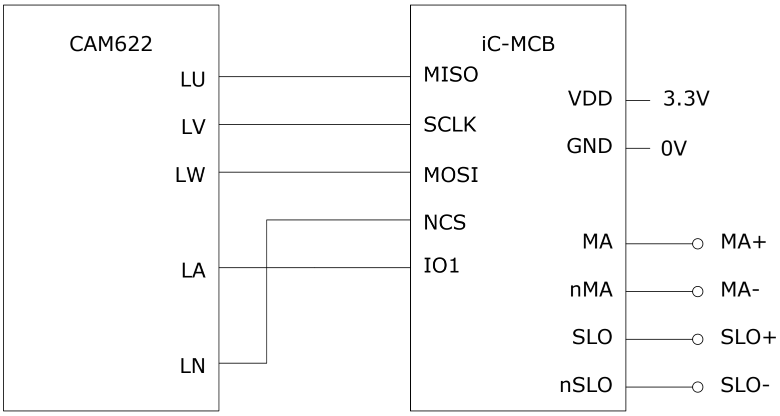


Figure 32 Connecting an external iC-MCB for BiSS (3.3V level SLO)

Note that the iC-MCB must be powered from the same 3.3V supply as the CAM622. Powering the iC-MCB from 5V is not possible since in this case its logic thresholds are not guaranteed to work with a 3.3V level signals. When powered from a 3.3V supply like this, the iC-MCB's SLO and nSLO outputs swing between 0V and 3.3V, not the 5V levels typically associated with RS422.

Figure 33 shows an alternative arrangement using external line transceivers. This circuitry supports both encoder ABN outputs and BiSS. This time the SLO+ and SLO- outputs swing between 0V and 5V. In this case the CAM622 is designed to work with THVD1420 line transceivers. They could possibly be substituted with alternative line transceivers, but these alternatives must have identical controls.

The A+ and A- lines are outputs when used for quadrature ABN (A1 high), and they are the MA+ and MA- BiSS inputs when used for BiSS (A1 low). The B+ and B- lines are outputs for both quadrature ABN and for BiSS (A0 high in both cases). When used for BiSS they are referred to as SLO+ and SLO-. N+ and N- are not used for BiSS. Together the A+ (MA+), A- (MA-), B+ (SLO+), B- (SLO-), N+ and N- lines are referred to as "ABN/BiSS signal lines".

Table 46 lists the settings required for each circuit option.

Table 46 Settings required for each BiSS circuit option

Setting	BiSS only circuit of Figure 32	BiSS and ABN circuit of Figure 33
BISSEN	1	1
ABEN	0	1
A0	User configurable, sets state of A0 pin	1
A1	User configurable, sets state of A1 pin	1
A0CTRL	0	1
A1CTRL	0	1
ESE	0	1
CB_SLO	0	1

Decoupling capacitors must be added according to the manufacturer's datasheets.

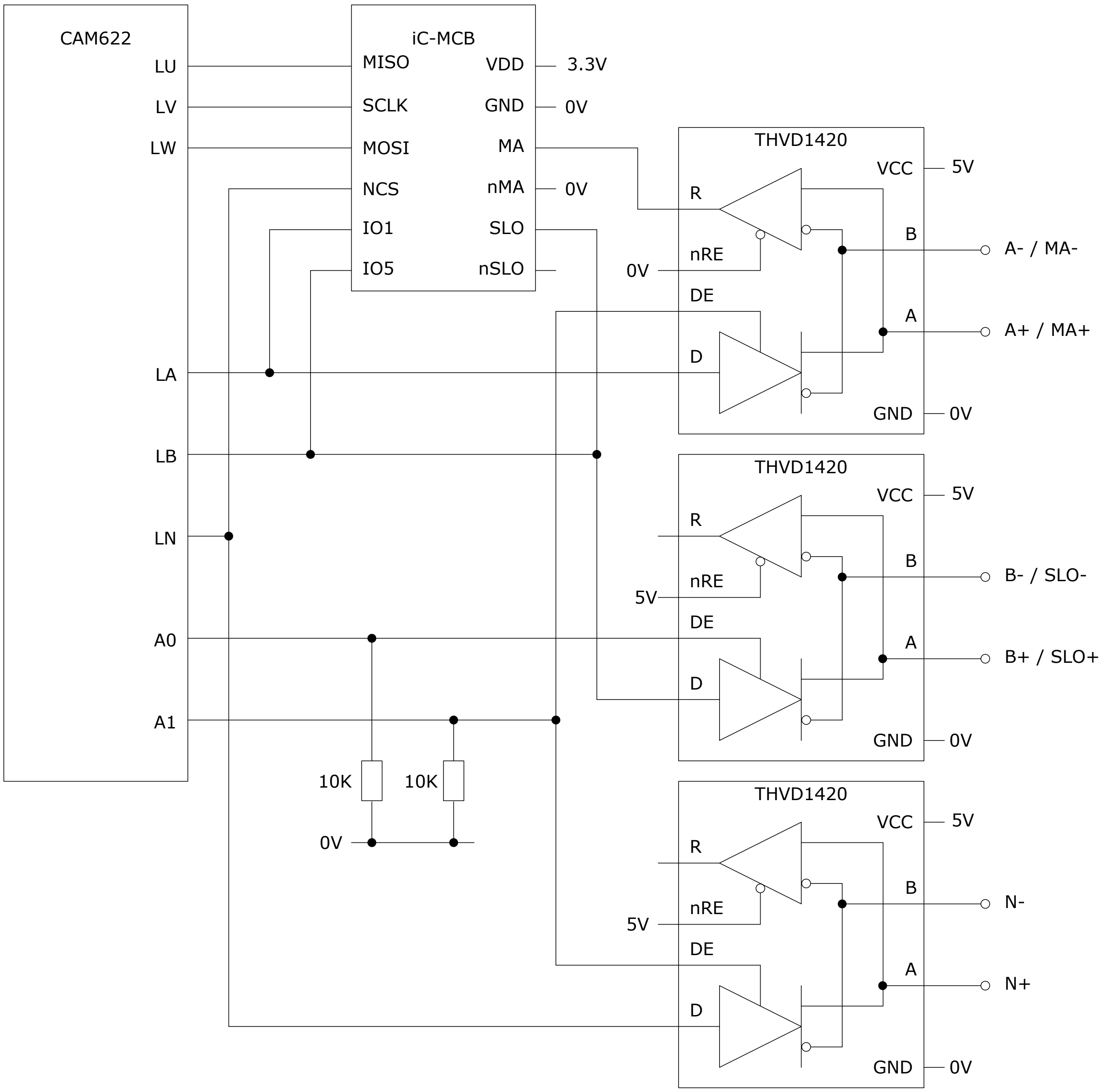


Figure 33 Connecting an external iC-MCB and line transceivers for BiSS and ABN

10.4 Use of BiSS with ABN

As noted above, the circuit of Figure 33 supports both BiSS and ABN communication. This allows a customer to use the same circuitry for devices that communicate over ABN and BiSS, avoiding the need for different hardware variants. Note that BiSS can be used to configure ABN operation, for example the number of AB cycles per revolution ABCYC. This means that an SPI connection is not required, except initially to enable BiSS and ABN. Subsequent updates to the CAM622's configuration can be done over BiSS. This means that a host can configure products over the ABN/BiSS signal lines even when there are no SPI connections, for example because they are inaccessible inside a product's housing and/or potted.

Even if a customer's applications do not require BiSS, it may still be attractive to support BiSS so that it can be used to configure a customer's products over the ABN signals without the need to expose SPI.

To allow both BiSS and ABN interface options, a product that includes the circuitry of Figure 33 it must initially be configured over the CAM622's SPI interface with ABEN=BISSEN=1. This can be done by making a temporary SPI connection to a host device. Please see Figure 20 for the connections required. A convenient host device is CambridgeIC's MultiComms Adapter, connected to a PC running CambridgeIC's CAM622 Configuration Tool. Please refer to the CambridgeIC CAM622 Software user manual for more details.

Having written ABEN=BISSEN=1 to the CAM622's non-volatile memory, subsequent configuration may be performed over the ABN/BiSS signal lines.

In order for the circuitry of Figure 33 to support both BiSS and ABN interfaces, the CAM622 includes a "ABN/BiSS Selection Process" illustrated in Figure 34.

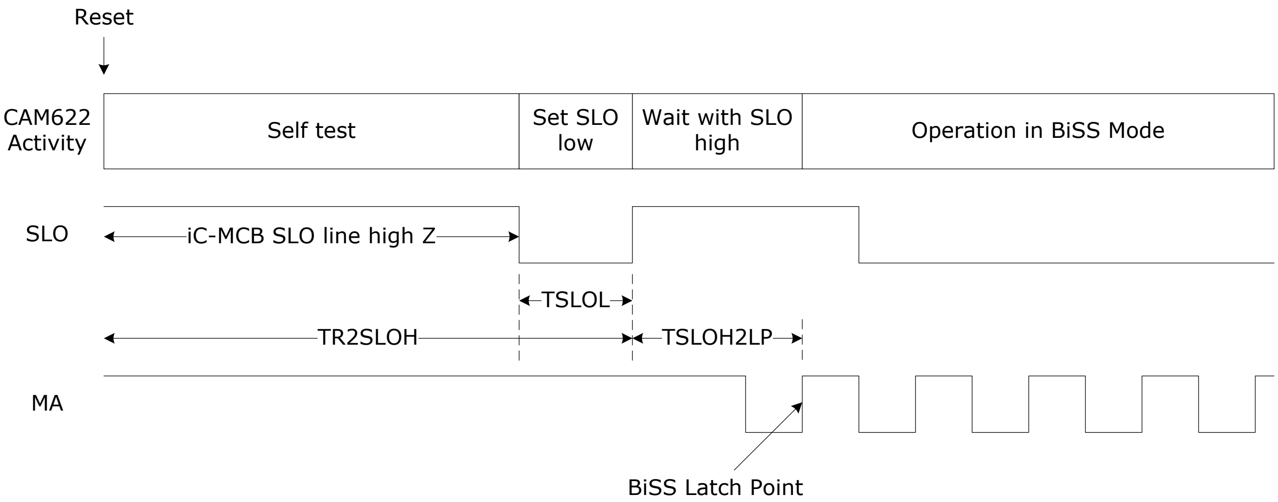


Figure 34 ABN/BiSS Selection Process

When the CAM622 comes out of a power on reset it first conducts a self test procedure. During this procedure the iC-MCB's SLO output is high impedance, which will be interpreted as a logic high by the connected THVD1420 line transceiver. If the CAM622's non-volatile configuration includes ABEN=BISSEN=1 then the CAM622 will then configure the iC-MCB, causing its SLO line to go low and then high again. The CAM622 starts a timer and checks for activity on MA. If a BiSS Latch Point is detected (see Figure 35) then the CAM622 will operate in BiSS mode until the next reset. If no latch point is detected within a maximum time TSLOH2LP(max) then the CAM622 will operate in ABN mode instead.

Table 47 ABN/BiSS Selection Process timings

Parameter	Description	Value
TR2SLOH(max)	Maximum time for CAM622 to make SLO go low then high	TR2nCSL from Table 11
TSLOL	Duration of SLO low period following reset	70µs approximately
TSLOH2LP(max)	Maximum time between SLO high edge and BiSS Latch Point to enter BiSS mode	80µs

When the ABN/BiSS Selection Process results in BiSS operation, the CAM622 clears its START bit. A Set/Clear START command (section 10.8) is needed to subsequently start measurements so that the system will report position values over BiSS. This helps protect a locked device against being used with BiSS unless the manufacturer specifically configures it for BiSS only, with ABEN=0.

Note that this ABN/BiSS Selection Process will not generally be supported by typical BiSS hosts, since it requires the BiSS master to respond quickly to activity on SLO following a reset. BiSS masters could be designed to support the ABN/BiSS Selection Process. However it is anticipated that most customers will use CambridgeIC's MultiComms Adapter for configuration over the ABN/BiSS signal lines when ABEN=BISSEN=1. In this case CambridgeIC's CAM622 Configuration Tool can be used for configuration, including:

- Basic configuration of the CAM622 including Motion Filter settings and SubType.
- Configuring ABN operation, for example the number of AB cycles ABCYC, or
- Configuring ABEN=0 for communication with a typical BiSS host, removing the ABN/BiSS Selection Process.
- Unlocking and locking the CAM622 for BiSS operation as necessary (section 10.11)

10.5 BiSS Timings

The BiSS Master starts a BiSS Frame with a falling edge on MA. The next rising edge of MA is the BiSS Latch Point. When delay compensation is active (DDC bit set to 0, INTERVAL set to 300) the CAM622 times its Latch Point Compensation to the BiSS Latch Point, so that the system "captures" a position value at exactly this point in time. Please see section 13.6 for more details.

Once Latch Point Compensation is complete, the CAM622 transmits data to the iC-MCB. The iC-MCB then prepares to transmit BiSS data. Once ready, it transmits the Start Bit on the next rising edge of MA. The timing parameter t_busy is the time between the Latch Point and when the iC-MCB is ready to clock out the Start Bit. Note that there will be an additional delay waiting for the next rising edge of MA, whose maximum value depends on the MA clock frequency.

This process and its timings are illustrated in Figure 35, and limiting values are given in Table 48. Please refer to the iC-MCB datasheet for more details of the iC-MCB's internal timings.

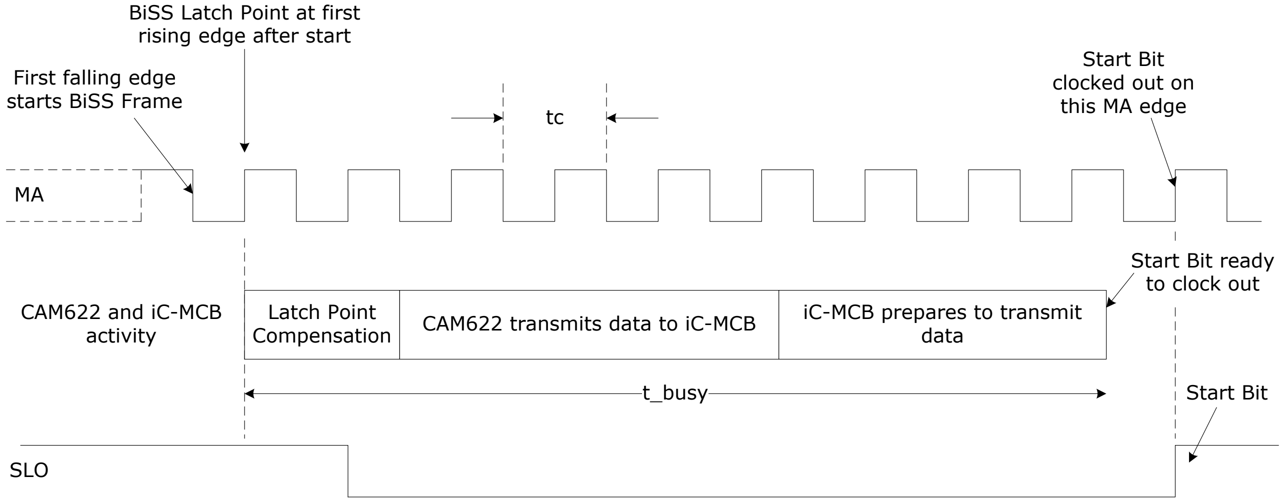


Figure 35 Start of BiSS Frame

Table 48 BiSS Timings

Parameter	**Value**	**Comments**
Maximum MA clock frequency	10MHz	= 1/(clock period tc)
Minimum BiSS Cycle Time	16μs	Interval between successive position values
Maximum t_busy between Latch Point and ready to send Start Bit	11μs	

10.6 BiSS Position Data Format

Each BiSS Frame the host clocks out data formatted as shown in Figure 36.

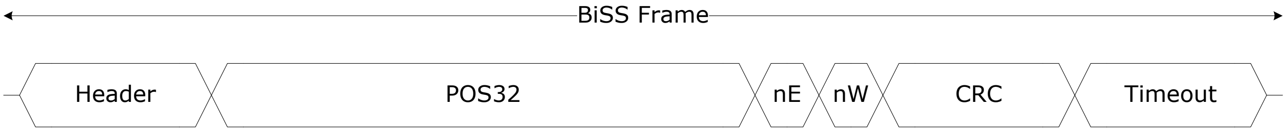


Figure 36 BiSS Data Format, BISSFORMAT=0

The Header includes the START bit illustrated in Figure 35, and a CDS bit used for Control Communication (section 10.7).

32 bits of position data ("POS32") then follows. When DDC=0, POS32 represents position at the exact time of the BiSS Latch Point, see section 10.5. Please refer to section 7.5 for the formatting of POS32.

The nE bit is an active low error bit. The CAM622 sets nE equal to the VALID bit reported over SPI. When it comes out of reset with the START bit set to 1, the CAM622 takes measurements reporting VALID=nE=0 until it detects an in-range target and the Motion Filter has initialised (section 13.1). Then the CAM622 will report VALID=nE=1, unless the target goes out of range. The CAM622 is not specifically designed to detect and report errors, but VALID=nE=0 can also be an indication of an error if it occurs beyond the initial start-up process and the target's absence is unexpected.

The nW bit is an active low warning bit. It may be configured as a low signal level warning, activating when Amplitude is less than LEDTHRESHOLD, see section 6.8. In this case the LEDTHRESHOLD setting must be configured to reflect a minimum value of AmplitudeA considered healthy, which depends on the sensor, its environment and the requirements of the application.

A 6-bit Cyclic Redundancy Check (CRC) of the POS32, nE and nW bits follows. The CRC polynomial is summarised in Equation 21:

Equation 21

$$CRC\ Polynomial: X^6 + X^1 + X^0\ (\text{"}0x43\text{"})$$
$$CRC\ Start\ Value: 0x00$$

10.7 BiSS Register Access

Each BiSS Frame described in section 10.6 transmits one additional bit of information from the BiSS host to the CAM622 ("CDM") and one bit from the CAM622 to the BiSS host ("CDS"). This is called Control Communication in BiSS terminology. Since only one bit of information is transferred per BiSS Frame, Control Communication is relatively slow.

Nevertheless over the course of many BiSS Frames it allows a BiSS host to access the CAMN622's diagnostic data (section 10.9) and to read and make changes to CAM622 configuration (section 10.10). To do these things, the BiSS host accesses BiSS Registers.

BiSS Registers must not be confused with the CAM622's registers listed in section 6.2. In this document "register" or "CAM622 register" will be used to refer to the CAM622's registers, to distinguish them from BiSS Registers.

There are two categories of BiSS Registers: BiSS Unbanked Registers and BiSS Banked Registers.

BiSS Unbanked Registers have addresses from 0x40 to 0x7F and are always available when performing register accesses. Their function is summarised in Table 49. Each address contains one byte.

Table 49 BiSS Unbanked Registers

Address	Definition	Access
0x40	Bank selection for banked registers, as defined in the BiSS specification	RW
0x41	Location of EDS bank, as defined in the BiSS specification. Fixed at 0 to indicate no EDS.	R
0x42…0x43	Profile ID, as defined in the BiSS specification	R
0x44…0x47	Serial number, as defined in the BiSS specification and configured in CAM622 configuration registers BISSSNUM0..BISSSNUM1.	R
0x48…0x4F	Reserved	
0x50…0x052	DATA3..DATA1 parameters for CAM622 commands via BiSS-C. See section 10.8.	RW
0x53	COMMAND ID for CAM622 commands via BiSS. See section 10.8.	RW
0x54	CMD_STATUS status value for CAM622 commands via BiSS. See section 10.8.	R
0x55…0x5F	Reserved	
0x60…0x6F	iC-MCB configuration (reserved, read-only???? – check with Alistair)	
0x70…0x71	CAMID mirrors the value of the CAM622 CAMID register.	R
0x72…0x73	SYSVER mirrors the value of the CAM622 SYSVER register	R
0x74…0x75	BOOTVER mirrors the value of the CAM622 BOOTVER register	R
0x76…0x77	SEVER mirrors the value of the CAM622 SEVER register	R
0x78…0x7D	DEVICE ID, as defined in the BiSS specification and configured in CAM622 configuration registers BISSDID0..BISSDID2	R
0x7E…0x7F	MANUFACTURER ID, as defined in the BiSS specification and configured in CAM622 configuration register BISSMID	R

BiSS Banked Registers are accessed from addresses from 0x00 to 0x3F. There are multiple banks, selected by the value of Bank Number written to BiSS Unbanked Register address 0x40. This selects between the BiSS Register Banks listed in Table 50.

Table 50 BiSS Register Banks

Bank Number	Start Address	Description	Section Ref
0x00	0x00	Do not use	
	0x20	Latched Measurement Results	10.9
0x01		CAM622 Configuration Banks	10.10
0x02			
0x03…0xFF		Reserved	

Each BiSS Bank holds up to 64 bytes of data, with addresses from 0x00 to 0x3F.

Please refer to the BiSS C Protocol Description for details of how to write to and read back from BiSS Registers over BiSS.

10.8 BiSS Related Commands

The BiSS host can command the CAM622 to take actions needed for access to the CAM622's measurement results and configuration. Each command comprises a Command ID (Table 52) and data. To initiate a command, the host first writes the required data bytes DATA3, DATA2 and DATA1 (together DATA[23:0]) to BiSS Unbanked Register addresses 0x50, 0x51 and 0x52 respectively. Then it writes the Command ID to BiSS Unbanked Register addresses 0x53. The command will take effect at the end of this write. The BiSS host can then read back a Command Status from register 0x54.

Table 51 Commands

Command ID	Description	Use	Survives Reset?
0x00	Apply Configuration	Transfers settings from CAM622 Configuration Banks to CAM622 registers.	N
0x01	Set/Clear START	Set DATA[23:0]=1 to set the CAM622 START bit to start measurements, 0 to stop.	N
0x02	Write Configuration to Flash	Transfers settings from CAM622 Configuration Banks to CAM622 registers. Set DATA[23:0]=1 to set the CAM622 START bit to start measurements, 0 to stop. If successful the CAM622 will reset and commence operation with new configuration.	Y
0x03	Latch Measurement Results	Creates a Latched Measurement Results Copy from the CAM622's current CTRL Register (Table 12) and Results Registers (Table 13).	N
0x04…0x0B	Reserved		
0x0C	Temporary Unlock Device	Set DATA[23:0] to a key value to attempt to unlock the CAM622 using that key, see section 10.11.	N
0x0D…0x0E	Reserved		
0x0F	Reset CAM622	Triggers the CAM622 to reset, returning to configuration stored in non-volatile memory.	N

The CAM622 returns the status of the most recent command CMD_STATUS in BiSS Unbanked Register address 0x54. Possible results are shown in Table 52. These values are only available immediately after a BiSS Command or another read of CMD_STATUS.

Table 52 Command Status

Value	Meaning	Comments
0x00	IDLE	No command has been actioned
0x01	ACTIVE	The last command is still being actioned
0x10	SUCCESS	Signals successful completion of last command, except when the command resulted in a reset.
0xE0	INVALID PARAMETER	Command ID was valid but DATA[23:0] is not valid for this command. For example the command Set/Clear START with DATA[23:0]=2
0xE1	ERROR, LOCKED	Results from any command not allowed when in the Locked state (section 10.11).
0xE2…0xEF	ERROR	
0xF0	NOT_FOUND	The Command ID is not recognised.
All other values	RESERVED	

10.9 Access to Diagnostic Measurement Results

The CAM622's primary measurement data is position, and this is reported every BiSS Frame as detailed in section 10.6.

In addition, the CAM622 takes measurements helpful for diagnostic and test purposes including signal Amplitudes and resonator frequency. These are reported in the CTRL register (Table 12), Results Registers (Table 13) and INFO registers (Table 14). When addressed over SPI their addresses are as shown in the Register Address column of Table 53.

A BiSS host can also read these register values over BiSS. To capture a complete synchronized set of diagnostic data, first use the Latch Measurement Results command (Table 51). Then read Latched Measurement Results from BiSS Register Bank 0x00, arranged as in Table 53:

Table 53 Latched Measurement Results

BiSS Bank 0x0 Address	Register Address	Register Name	Function
0x20, 0x21	0x00	CTRL	Read back value not relevant to BiSS
0x22…0x25	0x01, 0x02	POS32	Position measurement
0x26…0x29	0x03, 0x04	VELI32	Velocity measurement
0x2A, 0x2B	0x05	AMPA	Amplitude measurement, fine coils
0x2C, 0x2D	0x06	AMPB	Amplitude measurement, coarse coils
0x2E, 0x2F	0x07	BAMISMATCH	Position mismatch between coarse and fine coils
0x30, 0x31	0x08	FNUM	Resonator frequency measurement
0x32, 0x33	0x09	FILTLVL	Current value of Motion Filter level setting
0x34, 0x35	0x0A	CRC	Not relevant to BiSS
0x36…0x3D	0x0B…0x0E	INFO0…INFO3	Can be configured to read position in AB counts, see section 9.11.

Note that the position measurement POS32 appears in both the Latched Measurement Results and as the main process data reported each BiSS Frame. Most applications will use the POS32 value reported each BiSS Frame, because Latched Measurement Results are only accessible at a far slower rate. However the value reported in the Latched Measurement Results will be synchronised with the other data such as AMPA, which is helpful for some diagnostic purposes.

It typically takes 31 BiSS Frames to latch measurement results and 439 BiSS Frames to read a complete set of the registers listed in Table 53. If BiSS Frames repeat at 10kHz then this means it takes about 20ms to capture each set. The BiSS host can read Latched Measurement Results repeatedly, each time latching and reading Latched Measurement Results.

10.10 Reading and Writing CAM622 Configuration over BiSS

The CAM622 includes registers for configuring its operation, listed in section 6.2. Some are for basic configuration of the Sensing Engine, for example SECONFIG (section 6.4) includes SUBTYPE which tells the CAM622 what Subtype of sensor it is connected to. Others configure the CAM622 to communicate over a specific interface, for example the registers listed in Table 43 configure ABN operation, while those listed in Table 45 configure BiSS operation itself. All of these registers are accessible over BiSS, unless protected with a key (section 10.11).

A BiSS host accesses configuration registers from BiSS Register Banks 0x01 and 0x02. Table 54 summarises how addresses within these BiSS Banks correspond to CAM622 registers.

Note that CAM622 registers at addresses from 0x00 to 0x0F are not accessible by this mechanism. They correspond to registers CTRL including the START bit, the Results Registers and INFO registers. The START bit is controlled over BiSS, for example using the Set/Clear START command. Results and INFO registers are accessible using the mechanism described in section 10.9.

Table 54 Arrangement of CAM622 Configuration Banks

BiSS Bank	BiSS Bank Address	CAM622 Register Address	Register Name
0x01	0x00, 0x01	0x10	SECONFIG
	…	…	…
	0x3E, 0x3F	0x2F	UVWCOUNTL
0x02	0x00, 0x01	0x30	UVWCONFIG
	…	…	…
	0x1F	0x3F	BISSSNUM0

The BiSS host can simply read register values directly from the relevant bank and address(es). No command is necessary.

To write the CAM622's configuration, first perform BiSS writes to the relevant register addresses in banks 0x01 and 0x02. There are then two options: a temporary write to volatile memory or a non-volatile write to FLASH memory.

To temporarily configure the CAM622 with these new settings use the Apply Configuration command (Table 51). In this case the settings will only be stored in volatile memory and will be lost at the next reset. Since the START bit is not accessible over BiSS in this way, there is a separate Set/Clear START command that may be used to subsequently start and stop measurements.

Alternatively to save to non-volatile memory use the Write Configuration to Flash command. This command takes the state of the START bit as an argument. This START state will also be set in a non-volatile way. First set DATA[23:0] to the desired state of START, then initiate the Write Configuration to Flash command. If successful the CAM622 will reset itself. It will then come out of reset, perform self test, apply settings from non-volatile memory and start taking measurements continuously if START was set to 1, otherwise if it was set to 0 the CAM622 will remain in an idle state.

To support this process a BiSS host must be specially programmed. For simplicity, CambridgeIC's MultiComms Adapter can instead act as a BiSS host and the CAM622 Configuration Tool running on a PC can be used to configure CAM622 settings. Note that if the CAM622 is to subsequently communicate with a typical BiSS host without special programming, ABEN must be set to 0 to avoid the need for this host to perform the ABN/BiSS Selection Process detailed in 10.4.

10.11 Locking Configuration with a Key

The circuit of Figure 33 allows the CAM622 to generate ABN signals autonomously, while also being configurable over those same signal lines using BiSS. This configuration is typically done at manufacture or by one of the manufacturer's sales partners prior to sale to an end user. Configuration is also possible over SPI, however the advantage of configuring over BiSS is that no special provision is needed to connect to the SPI lines. The CAM622's SPI lines may be inaccessible due to enclosure and/or potting.

Access to the CAM622's commands (section 10.8), diagnostic data (section 10.9) and configuration (section 10.10) can be limited by locking the device. This can be used to prevent an end user from accessing and changing configuration. Typical uses are to prevent a user from changing:

- the value of ABCYC from the one intended by the manufacturer or their sales partner, and/or
- ABEN from 1 to 0, turning a product intended to generate ABN into one that supports BiSS without the special ABN/BiSS Selection Process detailed in section 10.4.

The factory default state of BISSKEY is 0x000000. In this state the CAM622 is always unlocked and a BiSS master has full access to configuration and commands. A non-zero value of BISSKEY must be used to lock the device.

To lock a CAM622, choose a secret non-zero value of BISSKEY, and write it to BISSKEY0 and BISSKEY1 registers together with KEYCODE set to 0xC0. When configuring for standalone operation over SPI (section 8), this write operation should be included in the intended configuration, which is a list of all registers to be configured and saved in non-volatile memory.

To unlock a CAM622 over BiSS, use the Temporary Unlock Device command with the correct BISSKEY value. If an incorrect key is used the CAM622 will enter the locked state, even if it was unlocked, and the command status will reflect this.

To check whether a CAM622 is locked, use the Temporary Unlock Device command with BISSKEY=0x000000. If the result is SUCCESS then the device is unlocked. If the result is ERROR, LOCKED then the device is currently locked and the Temporary Unlock Device command must be repeated with the correct value of BISSKEY to unlock the CAM622 until the next reset.

Table 55 shows what BiSS operations are possible when in the Unlocked state and Locked states.

Table 55 BiSS Operations Allowed by Lock State

BiSS Operation	Unlocked state	Locked state, ABEN=0	Locked state, ABEN=1
Read Serial Number, Device ID, Manufacturer ID from BiSS-C Unbanked registers	Allowed	Allowed	Allowed
Temporary Unlock Device command	Allowed	Allowed	Allowed
Reset CAM622 command	Allowed	Allowed	Allowed
Apply Configuration command	Allowed	Not allowed	Not allowed
Write Configuration to Flash command	Allowed	Not allowed	Not allowed
Set/Clear START command	Allowed	Allowed	Not allowed
Latch Latest Measurement Results command	Allowed	Allowed	Not allowed
Access to Diagnostic Measurement Results	Allowed	Allowed	Not allowed
Reading and Writing CAM622 Configuration	Allowed	Not allowed	Not allowed

When Locked, there are differences in access depending on the state of ABEN, as shown in Table 55. This allows the manufacturer to prevent an end user from configuring the CAM622 for BiSS when they intended it only for ABN operation. If ABEN=BISSEN=1 and the user manages to switch to BiSS operation following a reset (see ABN/BiSS Selection Process described in section 10.4), they will be unable to set the START bit over BiSS to take measurements. On the other hand if a manufacturer sets ABEN=0 then the CAM622 is intended for BiSS operation and not ABN, and the user can then start and stop BiSS measurements if they wish.

Note that when the CAM622 is unlocked, the BISSKEY value can also be updated over BiSS itself. Use the Temporary Unlock Device to unlock the CAM622, then write CAM622 configuration according to section 10.10 with the new value of BISSKEY. Care should be taken because it has the potential to render a device unusable.

To remove protection over SPI or BiSS use the same process described above for changing its value, except with BISSKEY set to 0x000000.

Note that a host connected over SPI has unrestricted access to the CAM622's configuration. The protection afforded by the locking process is only fully effective when access over SPI is prevented mechanically, for example by potting a module including the CAM622 and its SPI lines.

11 SENT Operation

11.1 SENT Introduction

The CAM622 includes a SENT interface for transmitting measurement data to a host device. It is a unidirectional interface that transmits data over a single conductor. SENT is typically used in automotive applications due to the low cost of implementation and robustness of data transmission.

This section assumes familiarity with SENT, and should be read in conjunction with SEA standard J2716 "Single Edge Nibble Transmission for Automotive Applications" ("J2716").

The SENT signal toggles between two states, high (near 5V) and low (near 0V). Figure 37 illustrates the transmission of a single SENT Message. It comprises a number of low periods of approximately equal duration TL. Data is encoded in the time between successive low going edges.

SENT data timings are measured in SENT Clock Ticks. The duration of a SENT Clock Tick may be varied depends on the needs of the application. According to J2716 the minimum SENT Clock Tick is 3μs, although the CAM622 can be configured for smaller values.

The SENT Message starts with a Synchronisation/Calibration Pulse with duration 56 SENT Clock Ticks. A host uses this Synchronisation Pulse to identify the beginning of a SENT Message. The host should measure the duration of this Synchronisation Pulse, dividing it by 56 to yield a value for the SENT Clock Tick that will be used to decode the following SENT Message data.

The time between each subsequent pulse and the previous one is used to transmit 4 bits of data (one nibble). A time of 12 times the SENT Clock Tick encodes the value 0x0, 13 times the SENT Clock Tick encodes the value 0x1 and so on, up to 27 times the SENT Clock Tick encodes the value 0xF.

The first nibble transmitted this way is the Status and Communication Nibble. There then follow between 2 and 6 Data Nibbles, followed by a checksum of the Data Nibbles ("FCHECKSUM[3:0]") defined in J2716.

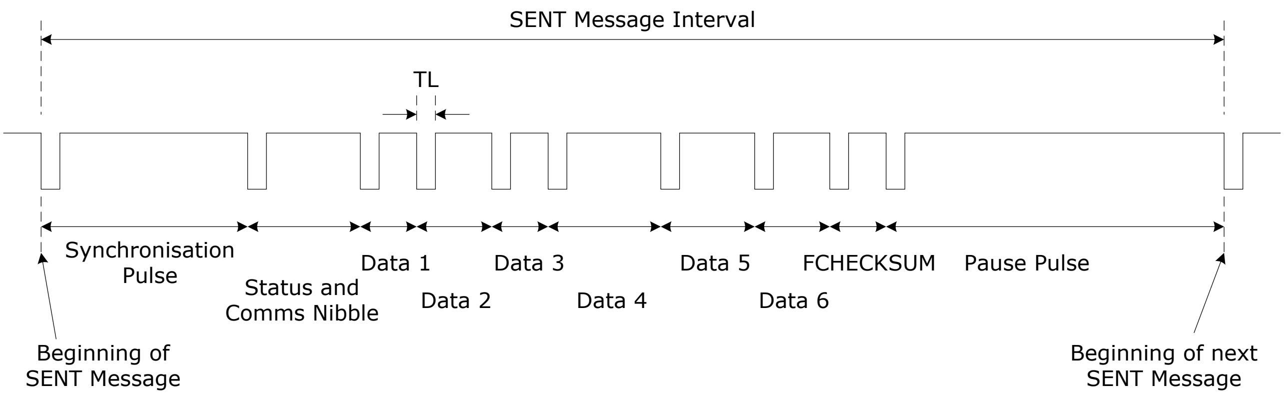


Figure 37 SENT Encoding Scheme, 6 Data Nibbles with Pause Pulse

The CAM622 then inserts a Pause Pulse before the next SENT Message begins. According to J2716 a Pause Pulse is optional. However the CAM622 always inserts one, so that the SENT Message Interval is constant and synchronised to the CAM622's Sample Interval.

The CAM622 generates SENT data on its LB pin. There are different options for how this is transmitted to a host device depending on the application's requirements, see section 11.4.

For applications where the CAM622 generates a SENT output, the CAM622 is configured at manufacture over its SPI interface. Please see section 8 for details of how to program a CAM622's configuration into its non-volatile memory over SPI, and section 11.2 for settings specific to SENT.

Once the CAM622 has been configured to generate SENT signals it will do so autonomously. It generates one SENT Message every SENTCOUNT CAM622 measurements. Each SENT Message includes a number of Data Nibbles encoding measured position and optionally velocity ("Fast Data") , see section 11.5. Each message also includes diagnostic data encoded in the Status and Communications Nibble using the Enhanced Serial Message Format defined in J2716, see section 11.6.

11.2 Settings Relevant to SENT

Table 56 Settings Relevant to SENT

Register	Field	Bits	Function	Ref
SENT CONFIG1	SENTEN	1	Set to 1 to enable SENT	6.15
	SENTPOL	1	Set to 0 for LB polarity equal to SENT signal shown in Figure 37, 1 for inverted to drive inverting circuitry such as Figure 38(c).	
	SENTOD	1	Set to 1 for LB open drain, 0 for push pull.	
	SENTFORMAT	4	Selects SENT Message data format, see section 11.5	
SENT CONFIG2	SENTTICK	8	Configures SENT Clock Tick length, see Equation 22.	
	SENTCOUNT	8	Configures SENT Message Interval, see Equation 23 or Equation 24	
SENTMSG1		16	Fixed values returned in diagnostic data that a customer can use for version numbers and serial numbers.	Table 23
SENTMSG2		16		
SENTMSG3		16		

Table 56 summarises the settings required to configure the CAM622's SENT interface. SENTTICK controls the SENT Clock Tick in units of 0.1µs, as shown in Equation 22. For example if SENTTICK=30 (0x1E) then the SENT Clock Tick length will be 3µs. The minimum value is specified in Table 57.

Equation 22

$$SENT\ Clock\ Tick/\mu s = SENTTICK \times 0.1\mu s$$

SENTCOUNT controls the SENT Message Interval, as shown in Equation 23. The INTERVAL register controls Sample Interval according to Equation 2, to yield Equation 24. For example if INTERVAL=300 and SENTCOUNT=33 (0x21) then the Sample Interval is 30µs and the SENT Message Interval is 30µs x 33 = 990µs.

Equation 23

$$SENT\ Message\ Interval = SENTCOUNT \times Sample\ Interval$$

Equation 24

$$SENT\ Message\ Interval/\mu s = SENTCOUNT \times INTERVAL \times 0.1\mu s$$

For SENT Messages with 6 data nibbles, the minimum allowed SENT Message Interval is 282 SENT Clock Ticks. This yields the inequality of Equation 25. The values of SENTCOUNT, SENTTICK and INTERVAL must be set to respect this inequality. The CAM622 does not check or enforce it.

Equation 25

$$SENTCOUNT \times INTERVAL > SENTTICK \times 282$$

When SENTOD=1 and SENTPOL=0, the CAM622's LB output is configured for open drain. It will be pulled low to generate a SENT signal low output, and will go high impedance when the SENT signal is high. When SENTOD=0, the LB output is driven both high and low. This is done with normal drive strength except during the Pause Pulse when it becomes a weak pull up or pull down as appropriate. The combination SENTOD=1 and SENTPOL=1 is not allowed.

11.3 SENT Specifications

Table 57 lists specifications relevant to the CAM622's SENT interface.

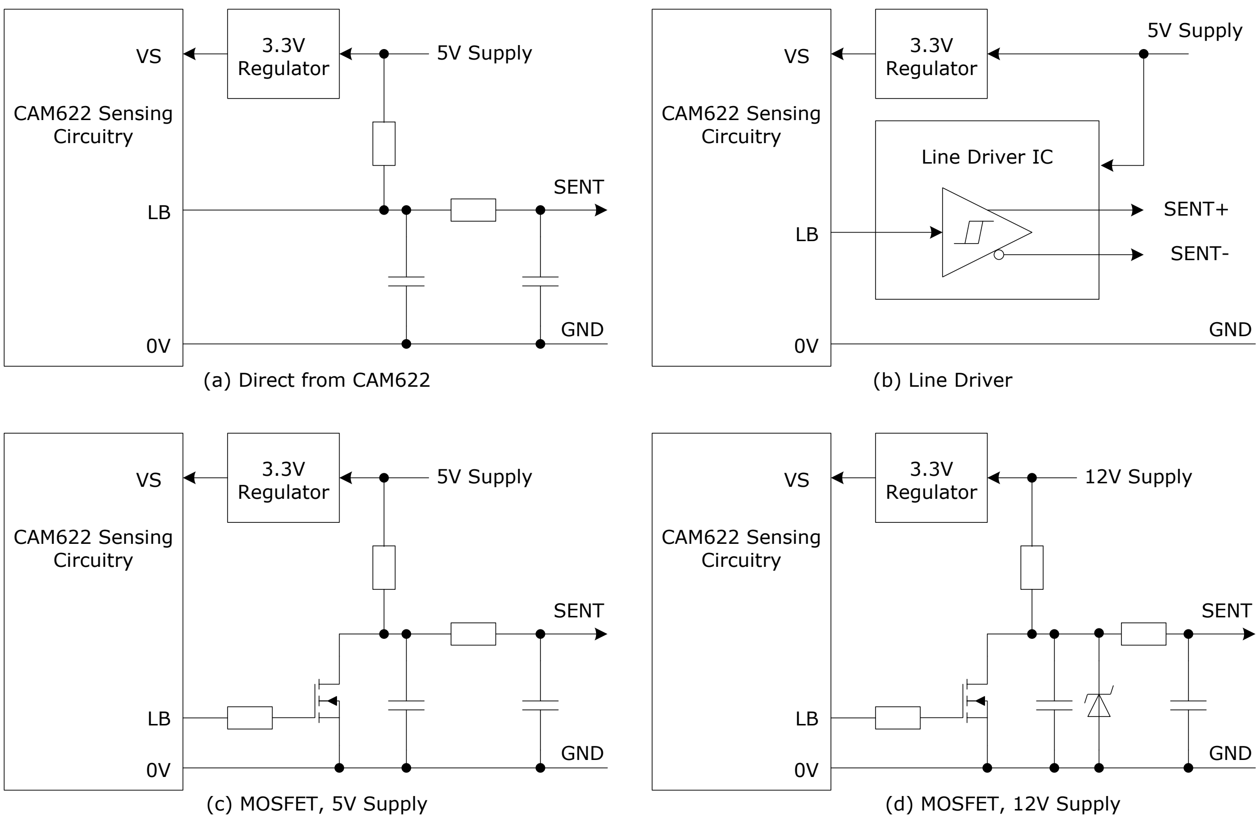
Table 57 SENT Specifications

Parameter	Description	Value	Comments
TL	SENT Signal low time	5 SENT Clock Ticks	Nominal
SENTTICK minimum value		1	Corresponds to 0.1µs
SENT Clock Tick tolerance		±3%	Across parts and Operating Temperature
Typical delay measured from actual position to Beginning of SENT Message		6µs	SENTFORMAT=0, DDC=0
		8µs	SENTFORMAT=1, DDC=0

11.4 Schematics for SENT Transmission

The CAM622's LB output can generate a SENT signal directly, as illustrated in Figure 38(a). The CAM622 runs from a 3.3V supply while the SENT logic high level is specified as 5V, so an external pull-up is used to generate a 5V logic high. LB is 5V tolerant. However higher voltages applied to LB could damage the CAM622, so this arrangement is only recommended where SENT signal connections are short and there is no scope for transients.

Figure 38(b) illustrates a more robust scheme using a 5V line driver. This arrangement is used on the CAM622 MultiComms Board, where line drivers are required to support other interfaces including ABN and BiSS. However it is not normal to transmit SENT using a line driver, because SENT signal edge speeds are excessive.



(a) Direct from CAM622

(b) Line Driver

(c) MOSFET, 5V Supply

(d) MOSFET, 12V Supply

Figure 38 SENT Transmission Schematics

Figure 38(c) and (d) illustrate the use of a MOSFET to transmit the SENT signal, with additional low-pass filter circuitry to slow down the SENT output. Figure 38(c) includes a pull up to 5V for generating a high SENT output. Where a 5V supply is not available the circuit of Figure 38(d) may be used, using a Zener diode to limit the high level SENT signal to 5V. Please refer to J2716 for SENT edge timing and signal strength requirements, to help select appropriate component values.

Table 58 lists SENT settings required for each SENT transmission approach.

Table 58 Settings required for each SENT Transmission Schematic

Schematic	Description	SENTPOL	SENTOD	Comments
Figure 38(a)	Direct from CAM622	0	1	CAM622 pulls LB low, external pull-up pulls it to 5V
Figure 38(b)	Line Driver	0	0	
Figure 38(c)	MOSFET, 5V Supply	1	0	External MOSFET inverts so need TXPOL=1
Figure 38(d)	MOSFET, 12V Supply	1	0	

11.5 Fast Data Transmission

Fast data refers to data transmitted as nibbles each SENT Message. Data is "Fast" because a complete set of data is transmitted each SENT Message. The CAM622 supports two different data formats.

When SENTFORMAT=0, the CAM622 transmits 24-bit position data using 6 Data Nibbles. These are the top 24 bits of the CAM622's POS32 measured position value defined in section 7.5. Table 59 details the location of each bit within the Data Nibbles.

Table 59 SENT Message, SENTFORMAT=0

SENT Nibble	Nibble Bit 3	Nibble Bit 2	Nibble Bit 1	Nibble Bit 0
Status and Communication	SCN3	SCN2	0	nVALID
Data 1	POS32[31]	POS32[30]	POS32[29]	POS32[28]
Data 2	POS32[27]	POS32[26]	POS32[25]	POS32[24]
Data 3	POS32[23]	POS32[22]	POS32[21]	POS32[20]
Data 4	POS32[19]	POS32[18]	POS32[17]	POS32[16]
Data 5	POS32[15]	POS32[14]	POS32[13]	POS32[12]
Data 6	POS32[11]	POS32[10]	POS32[9]	POS32[8]
Fast Data Checksum	FCHECKSUM[3]	FCHECKSUM[2]	FCHECKSUM[1]	FCHECKSUM[0]

The Status and Communication Nibble's bit 0 ("nVALID") indicates whether position data is valid (0) or not (1).

When SENTFORMAT=1, the CAM622 transmits 14-bit position data and 10-bit encoded velocity data ("LOG2V") using 6 Data Nibbles. This corresponds to J2716 SENT frame format H6. Under J2716, the nibbles of the second data channel LOG2V are reversed, resulting in the bit locations shown in Table 60.

Table 60 SENT Message, SENTFORMAT=1

SENT Nibble	Nibble Bit 3	Nibble Bit 2	Nibble Bit 1	Nibble Bit 0
Status and Communication	SCN3	SCN2	0	nVALID
Data 1	POS32[31]	POS32[30]	POS32[29]	POS32[28]
Data 2	POS32[27]	POS32[26]	POS32[25]	POS32[24]
Data 3	POS32[23]	POS32[22]	POS32[21]	POS32[20]
Data 4	POS32[19]	POS32[18]	LOG2V[1]	LOG2V[0]
Data 5	LOG2V[5]	LOG2V[4]	LOG2V[3]	LOG2V[2]
Data 6	LOG2V[9]	LOG2V[8]	LOG2V[7]	LOG2V[6]
Fast Data Checksum	FCHECKSUM[3]	FCHECKSUM[2]	FCHECKSUM[1]	FCHECKSUM[0]

The velocity encoding scheme is designed for slow rotating applications including vehicle steering, up to ±2200°/s. Velocity is logarithmically encoded to offer high resolutions at low speed and a large dynamic range of speeds. To convert LOG2V into velocity in degrees per second use Equation 26:

Equation 26

$$Velocity\ in\ ^\circ\ per\ second = \ sign(LOG2V) \times 0.03523\ \times 2^{m}$$
$$m = \ |LOG2V|\ \times 2^{-5}$$

Equation 17 is only valid when INTERVAL is set to 300, to yield a nominal Measurement Interval of 30µs.

The code LOG2V=0 is reserved to indicate an invalid measurement reading, which can occur during start-up and if the target is out of range. The Status and Communication Nibble also includes an nVALID bit. However this nibble is not included in the checksum FCHECKSUM, so for reliability it is strongly recommended to treat LOG2V=0 as the main indication of valid measurement data.

11.6 Diagnostic Data Transmission

Fast Data Transmission of position and optionally velocity information is detailed in section 11.5. This is optimised for a high update rate, by transmitting multiple SENT Nibbles per message.

Diagnostic data including Amplitude values, relative frequency and BA Position Mismatch are useful during development, for example to check the alignment of a target and the effect of any nearby metal. The values are also an important indicator of system health for manufacturing test. A high update rate is not as important for these values, and the CAM622 transmits them using the Enhanced Serial Message Format specified in J2716.

One 16 bit diagnostic data word is transmitted every 18 SENT messages, together with a 4-bit Message ID and 6-bit CRC checksum. These are encoded in the Status and Communication Nibble's SCN3 and SCN2 bits, as illustrated in Table 61. The 6 bit CRC checksum is calculated as defined in J2716.

Table 61 Serial data bit values by SENT Message number

SENT Message No	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
SCN2	6 bit CRC						Diagnostic Data [11:0]]											
SCN3	1	1	1	1	1	1	0	1	Message ID [3:0]				0	Diagnostic Data [15:12]				0

Note that the first 6 SENT messages in the sequence have SCN3 set to 1. The host uses this pattern to identify the beginning of the sequence.

The CAM622 transmits 16 words of diagnostic data, as specified in Table 62. It transmits each one sequentially and then repeats the sequence. Table 62 includes the Register Address for each word of diagnostic data within the CAM622's main register space addressable over SPI, see section 6.2.

Table 62 Diagnostic data for each Message ID

Message ID	**Contents**	**Register Address**	**Description**
0x0	POSH	0x01	Position values, usually ignored by a host because upper bits of position are transmitted faster each SENT Message.
0x1	POSL	0x02	
0x2	VELH	0x03	Velocity values, also available at higher speed in encoded form when SENTFORMAT=1.
0x3	VELL	0x04	
0x4	AMPA	0x05	Diagnostic data for development and manufacturing test
0x5	AMPB	0x06	
0x6	BAMISMATCH	0x07	
0x7	FNUM	0x08	
0x8	FILTLVL	0x09	
0x9	INFO0	0x0B	CAMID (0x0622 for "CAM622") (1)
0xA	INFO1	0x0C	SYSVER (Version of Application Code) (1)
0xB	INFO2	0x0D	BOOTVER (Version of Bootloader) (1)
0xC	INFO3	0x0E	SEVER (Version of Sensing Engine) (1)
0xD	SENTMSG1	0x44	Fixed values that a customer can use for version numbers and serial numbers.
0xE	SENTMSG2	0x45	
0xF	SENTMSG3	0x46	

Note (1): The values returned in the INFO registers are configurable using the INDEX10 and INDEX32 registers, see section 6.10. With the CAM622's default INDEX register values, the CAM622 returns the version numbers shown.

It takes 18 SENT Messages times 16 Message IDs to transmit one full set of diagnostic data, a total of 288 SENT Messages. If the SENT Message Interval is configured to a typical value of 1ms, this means about 3 sets per second.

12 Motor Commutation Operation

Motor commutation outputs are not currently implemented. This section is preliminary information, to advise customers of a possible future CAM622 enhancement.

The CAM622 may be configured to generate digital UVW outputs. These are for motor commutation. The UVW signals are like those generated by digital Hall devices commonly installed in brushless motors.

12.1 Settings Relevant to Commutation Signals

This section describes available settings for the UVW outputs. These are typically written to the CAM622's non-volatile memory in a configuration step at manufacture, see section 8. Once programmed in this way the CAM622 operates autonomously with the chosen settings.

Table 63 lists the available settings associated with the UVW outputs. The following subsections detail their effects.

Table 63 Commutation Settings

Register	Field	Bits	Function	Section Ref
UVW CONFIG	UVWEN	1	Set this bit to 1 to enable UVW signals.	6.12
	UVWDIR	1	Reverses UVW direction when set.	12.3
	2PH	1	Set to 0 for conventional 3-phase motor commutation, or 1 for 2-phase, see section 12.4.	12.4
UVWPOS		16	Controls the position of the start of the UVW signal cycle.	12.3
UVWCYC		6	Controls the number of UVW cycles around 360°. Set to equal motor pole pairs.	
INDEX32		16	Set to 0x1F1E so INFO2 and INFO3 contain UVWCOUNTH and UVWCOUNTL	6.10

The UVW outputs are also affected by the Motion Filter described in section 13, and its settings are listed in Table 64. These settings affect the dynamic behaviour of the position values fed into the communication edge calculation. Filtering can help reduce noise. This in turn can help control UVW edge flicker if needed.

The Motion Filter's DDC setting ("Disable Delay Compensation") controls whether the Motion Filter compensates for measurement and filter delays. It is also used to control delay compensation applied to the communication edge calculation. When set to the default value of 0, the CAM622 compensates for delays both in the Motion Filter itself and in UVW synthesis. When set to 1, these compensation is disabled, resulting in the delays specified in Table 65.

12.2 Interfacing to LU, LV and LW

The CAM622 chip's LU, LV and LW outputs are 3.3V nominal logic level and have a relatively low output voltage and current (section 2.4). These may be sufficient where they can be directly connected to a 3.3V powered host device on the same PCB or with short and direct connections. However some applications will require signal transmission over longer cable lengths, and typically require external MOSFETs as illustrated in Figure 39.

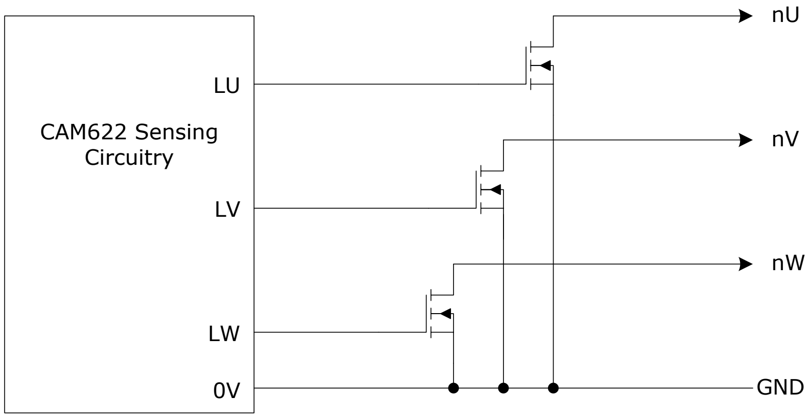


Figure 39 Generating open drain U, V, W signals using external MOSFETs

In this example the outputs nU, nV and nW will have the opposite polarity to LU, LV and LW. Note that there is no provision for the CAM622 to invert LU, LV and LW signals because inversion is equivalent to an electrical 180° shift in phase. This phase shift can be implemented with an appropriate change to the value of UVWPOS and hence UVWoffset (section 12.3). UVWoffset is in mechanical degrees, and an electrical phase change of 180° is equivalent to a mechanical angle change of ±180°/UVWCYC.

12.3 Effect of UVWDIR, UVWPOS and UVWCYC

Figure 40 illustrates how LU, LV and LW signals vary with position when enabled (UVWEN=1), and with 2PH=0. LU, LV and LW are 3-phase digital signals with UVWCYC repeats around a circle.

Figure 40(a) illustrates the signals when UVWDIR=0. In this case LU transitions high at the UVW zero position when moving in the positive angle direction. The LV signal lags LU, and the LW signal lags LV. Please refer to the Type B Sensor Reference Manual for a definition of the coordinate system used, including Actual Angle and its direction. Figure 40(b) illustrates the signals when UVWDIR=1. In this case the LU, LV and LW signals transition in the opposite direction.

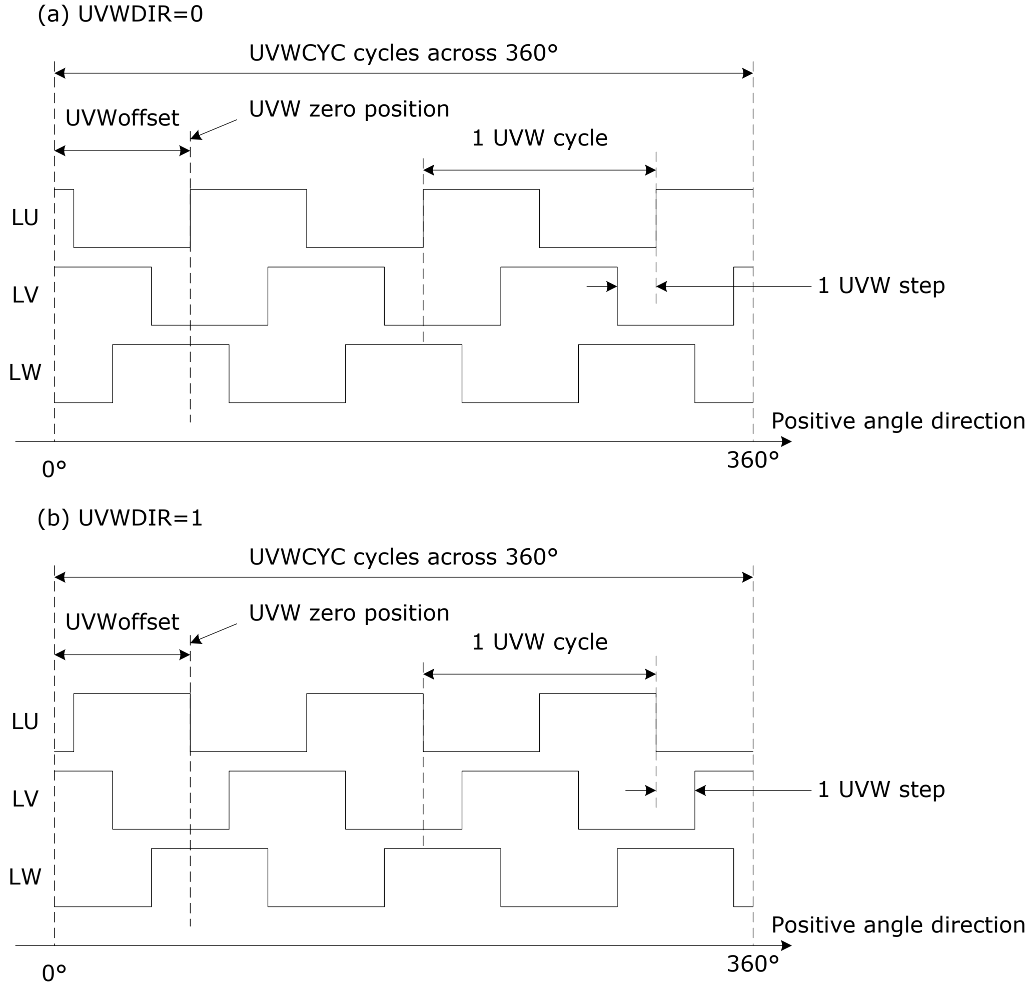


Figure 40 LU, LV and LW signals as a function of position, 2POL=0

The UVWCYC parameter controls the number of UVW cycles around 360°. This is marked in Figure 40, which illustrates the case when UVWCYC=8. Set UVWCYC to the number of cycles required. When used for motor commutation, UVWCYC is usually the number of motor pole pairs. UVWCYC must not be set to 0. The maximum value of UVWCYC is 63.

A UVW step is the smallest angle change that can be directly measured from UVW signals, and is marked in Figure 40.

The UVW zero position is variable across 360° and is controlled by the UVWPOS setting. UVWPOS is a 16-bit integer value. The relationship between UVWPOS and the physical angle UVW offset shown in Figure 40 is given by Equation 27.

Equation 27

$$UVWoffset = \frac{UVWPOS}{65536} \times 360°$$

Note that if UVWCYC is greater than 1 then there is more than one possible value of UVWPOS and UVWoffset that have the same effect. For example Figure 40 illustrates the case where UVWoffset=70°. The same waveforms would result if UVWoffset were set to 70° + 120° = 190° or 70° + 240° = 310°.

12.4 Effect of 2PH

When the 2PH bit is set to 1, the LU and LV signals behave as 2-phase digital signals, with LW always low. This is illustrated in Figure 41. The effects of UVWCYC and UVWPOS are unchanged.

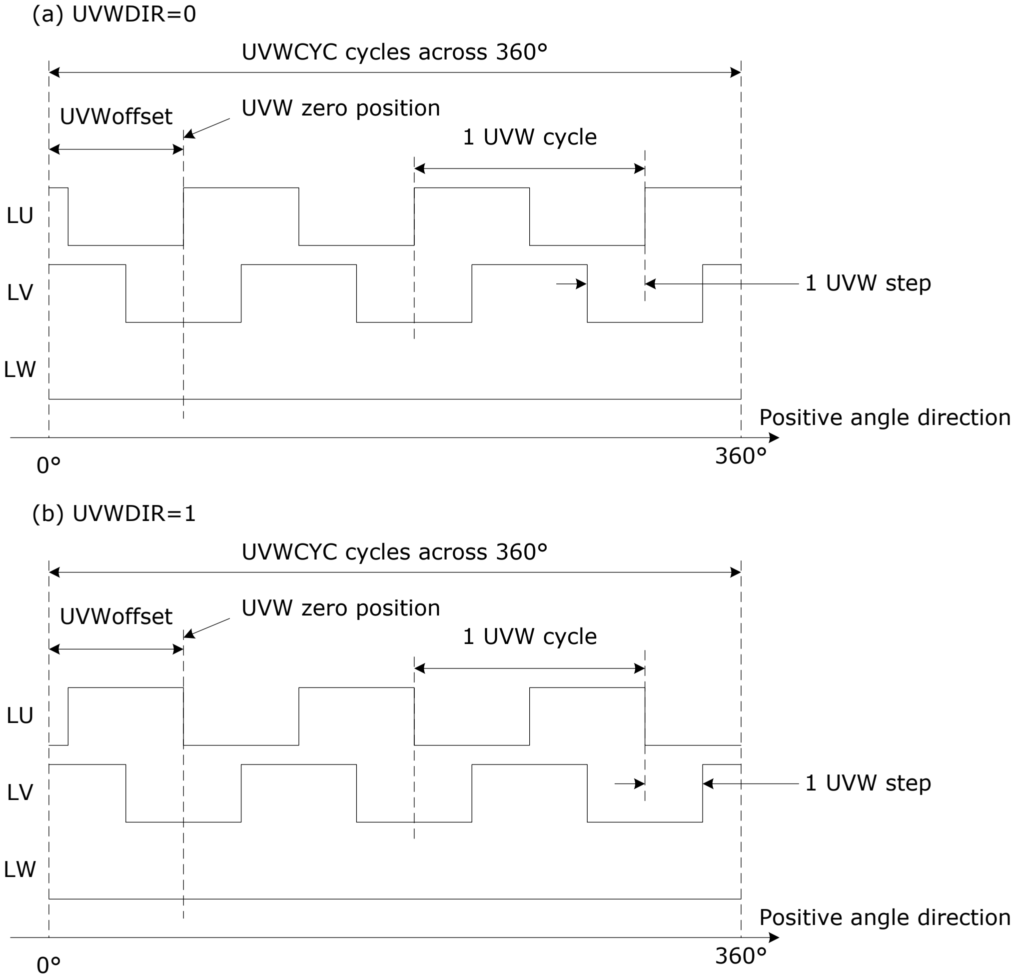


Figure 41 LU, LV and LW signals as a function of position, 2POL=1

Almost all brushless DC motors are 3-phase, and in this case 2PH must be set to 0. There are rare exceptions, including 2-phase stepper motors. 2-phase stepper motors are not usually operated as brushless motors with commutation, but in principle they can be using a CAM622 with 2POL=1.

12.5 Reading UVW State over SPI

When START=1 and UVW generation is enabled with UVWEN=1, the UVWCOUNTH register returns the UVW state number. This normally runs from 0 to 5 for commutating a 3-phase motor. Alternatively it runs from 0 to 3 for a 2-phase motor (when the 2PH bit is set). The value of UVWCOUNTH reflects the state of the LU, LV and LW pins.

The UVWCOUNTL register returns the fractional part of the state number. This value is only available through the SPI interface, and not on physical pins.

The UVWCOUNTH and UVWCOUNTL registers have addresses beyond 0x0E, so a host SPI device can not read them directly when START=1. Instead, they may be accessed through the Information Registers. When configured according to Table 63, ABCOUNTH will appear at address 0x0C (INFO2) and ABCOUNTL at 0x0D (INFO3).

To read ABCOUNTH and ABCOUNTL registers, perform an SPI read transaction up to and including INFO3. A delay of at least TnCSH_ABNread(min) is required after this SPI transaction and before the next one. This is specified in Table 11.

13 Motion Filter

The Sensing Engine includes a Motion filter block, see Figure 4. Its purpose is to reduce the noise present in raw position and velocity data, before this data is passed to the Interface Processor. A typical application is to increase the number of noise free encoder edges per revolution without adding unwanted hysteresis.

This increase in noise free resolution is at the expense of dynamic behaviour. A host will typically use the frequency of encoder edges to determine velocity, and the filter delays the CAM622's response to changes in velocity. This is only noticeable in applications with large velocity changes in a very short period of time. Please see section 13.4.

By default, the Motion Filter includes compensation for internal delays. This means that the position reported over the encoder, commutation and SPI interfaces is free from position lag when operating at constant velocity.

The Motion Filter can be configured to adapt the amount of filtering, to offer high resolution at constant velocity and a faster response to changes in velocity. Please refer to section 13.5.

The use of the Motion Filter is optional, and it can be disabled (section 13.7).

13.1 Motion Filter Design

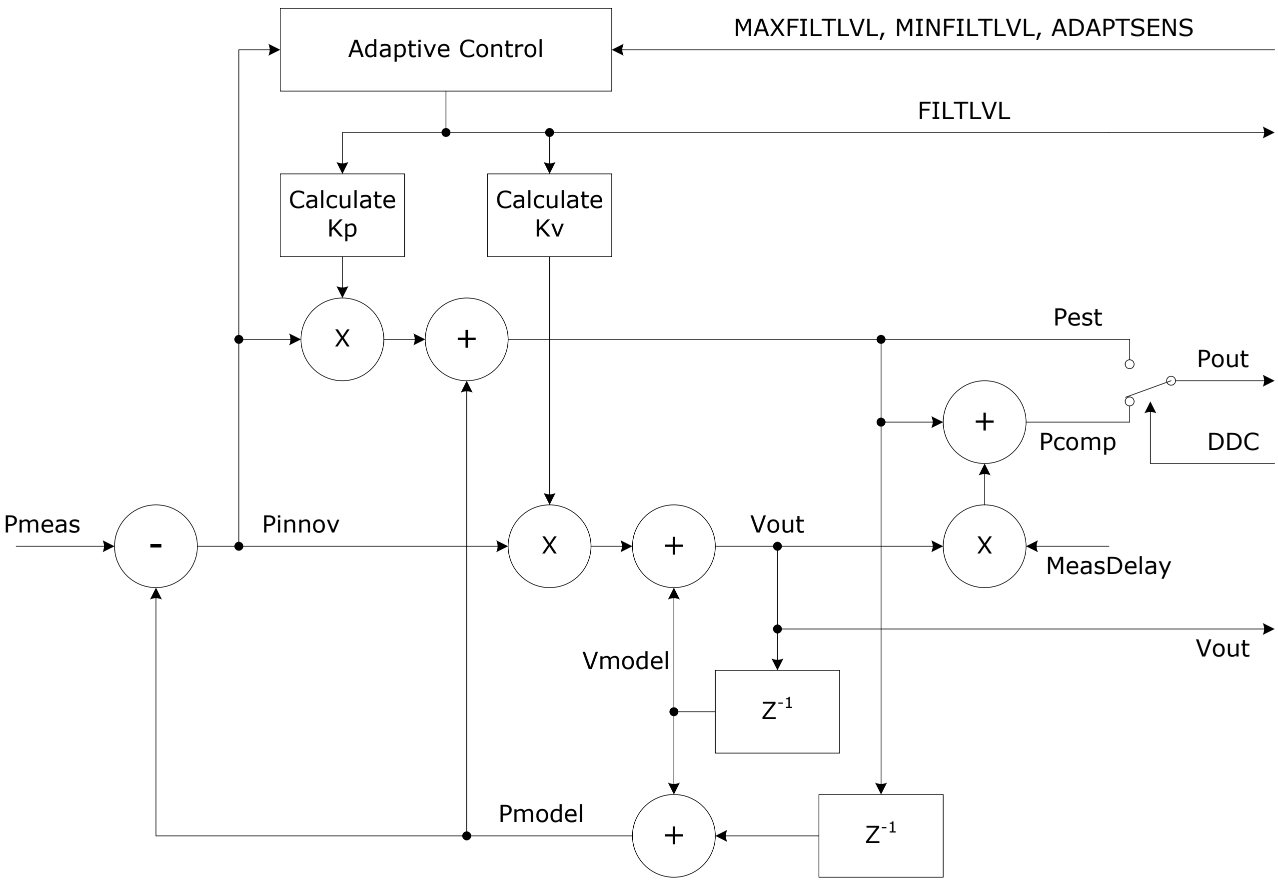


Figure 42 Motion Filter design

The Motion Filter is illustrated in Figure 42. Its input Pmeas comprises raw measurement samples from position calculations. Its primary output is filtered position Pout. It also outputs a velocity estimate Vout. For the purposes of Figure 42 the units of velocity are position per sample.

The nominal Sample Interval is 30µs (section 6.5). Incoming and outgoing measurements are updated each sample.

The Motion Filter's responsiveness is controlled by FILTLVL. FILTLVL can range from 0 (no filtering) to 255 (maximum filtering). FILTLVL is determined by the Adaptive Control block, which calculates an appropriate value based on host settings, apparent acceleration and motion history. The value of FILTLVL may be read over the CAM622's SPI interface, to assist with filter tuning.

The filter settings MAXFILTLVL, MINFILTLVL, ADAPTSENS and DDC are under host control. They are usually stored in non-volatile memory when the CAM622 is generating encoder and commutation signals autonomously. Their functions are summarised in Table 64.

Table 64 Motion Filter controls

Setting	Function	Comments
MAXFILTLVL	Sets maximum FILTLVL	
MINFILTLVL	Sets minimum FILTLVL	Must be set to MAXFILTLVL or less
ADAPTSENS	Sets Sensitivity to acceleration / noise	Set to 0 for no adaptive behaviour (FLTLVL=MAXFILTLVL), or see section 13.5 for how to configure adaptive filtering.
DDC	Disable Delay Compensation	1 to disable delay compensation, 0 otherwise

The Motion Filter's design is a type of Kalman filter. It includes a control loop comparing the raw measurement data Pmeas with Pmodel and generating a difference signal Pinnov which it attempts to minimise. Pmodel is the Motion Filter's prediction of Pmeas, based on the assumption that motion is at constant velocity (successive Pmeas values equally spaced). Pinnov ("innovation") expresses the amount of unpredictable change in Pmeas. This is a combination of measurement noise and acceleration. Measurement noise is by its nature unpredictable. The CAM622 does not "know" what acceleration to expect at any given time, and can only rely on raw measurements to detect it.

The Motion Filter adds a fraction Kp of Pinnov to Pmodel to determine its best estimate of position Pest. Kp is calculated from FILTLVL. When FILTLVL is small Kp is near 1, so Pest is weighted strongly in favour of raw data Pmeas. When FILTLVL is near its maximum Kp is near 0, so Pmeas is weighted in favour of Pmodel, and hence the history of past values.

Similarly the Motion Filter adds a fraction Kv of Pinnov to Vmodel to determine its best estimate of velocity Vest. Kv is also calculated from FILTLVL. Its value also range from near 1 (FILTLVL small) to near 0 (FILTLVL near maximum).

Kp and Kv are calculated in such a way that the filter's dynamic behaviour is always critically damped. Note that Kv is generally much smaller than Kp when FILTLVL is large, so that the Motion Filter's estimate of velocity Vout is more sluggish than its position output Pout.

Vmodel is the Motion Filter's prediction of Velocity for the current sample. The Motion Filter does not attempt to estimate acceleration, so Vmodel is simply equal to the previous sample's velocity estimate Vout. "Z^{-1}" blocks in Figure 42 denote a delay of one sample.

As noted above Pmodel is the Motion Filter's prediction of Pmeas based on the assumption that motion is at constant velocity. Pmodel is calculated from the previous estimated position (Pest delayed by one sample) plus Vmodel, the expected position change per sample.

The Sensing Engine's ADCs, detection and position calculation introduce a delay MeasDelay to Pmeas values relative to a snapshot of actual physical position. This same delay is present in Pest values. For the purpose of Figure 42 MeasDelay has units of a fraction of the Sample Interval. The motion filter can compensate for this delay by estimating the position lag it causes (Vout x MeasDelay) and adding this to Pest, to yield Pcomp. The position reported to the Interface Processor Pout is Pcomp by default, or Pest if DDC is set to 1. When the Interface processor receives a new value of Pout, it is nominally the instantaneous position of the target "right now" if DDC is set to 0. This means the measurement of Pout has a Phase Delay of zero, see section 13.3.

To synthesise encoder edges, the Interface Processor updates the AB edge frequency each sample and aims to achieve the correct count value when the next sample comes along. If DDC is set to 0, it predicts position at this next sample based on the current sample's Pout (=Pcomp) plus Vest. That is, it applies compensation for a further sample's worth of delay, over and above the compensation for MeasDelay shown in Figure 42. This is done to ensure that encoder edges occur at the appropriate physical position without lagging behind due to delays. This means the encoder edges also have a Phase Delay of zero, see section 13.3.

When the Sensing Engine first detects a target, it initialises the Motion Filter using a small value of FILTLVL. This increases on subsequent measurements until FILTLVL reaches MINFILTLVL, at which point the Sensing Engine will report VALID. The number of measurements this takes depends on the value of MAXFILTLVL. When MAXFILTLVL is 70 or less, filter initialisation takes up to 10 measurements. This increases up to a maximum of 80 measurements when MAXFILTLVL=255.

13.2 Resolution Improvement

The amount of filtering applied by the Motion Filter is controlled by FILTLVL. The host may select a fixed amount of filtering, by setting both MAXFILTLVL and MINFILTLVL to the desired FILTLVL value. Alternatively it may specify upper and lower bounds, MAXFILTLVL and MINFILTLVL respectively, and use adaptive filtering (section 13.5). Providing the sensitivity of adaptive filtering set by ADAPTSENS is set appropriately, FILTLVL will equal MAXFILTLVL when the target is stationary or rotating at constant velocity, and will reduce as low as MINFILTLVL when acceleration is detected.

Figure 43 shows how the value of FILTLVL influences the system's noise free resolution. The vertical axis is improvement in Noise Free Resolution relative to raw measurements taken with the Motion Filter disabled (section 13.7). The scale is in bits. One bit of Noise Free Resolution is equivalent to a factor of 2 reduction in position noise, for example as measured by its peak to peak value or standard deviation. 2 bits is therefore equivalent to a factor of 4 reduction in noise.

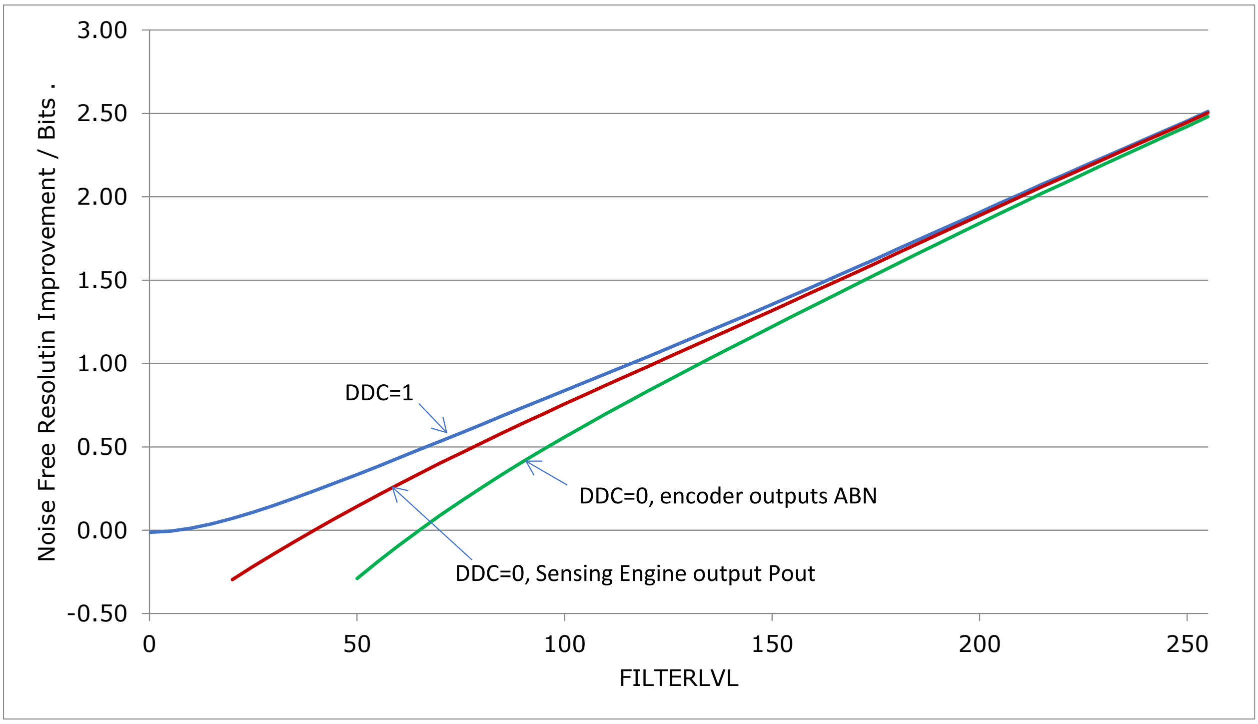


Figure 43 Noise Free Resolution Improvement as a function of FILTLVL

There are three plots in Figure 43. The "DDC=1" plot is with delay compensation disabled. This yields the best Noise Free Resolution, at the expense of Phase Delay (section 13.4).

There are two "DDC=0" plots. The "Sensing Engine output" plot concerns noise measured directly at the Motion Filter's output Pout, which is an input to the Interface Processor. It also applies to when the host device reads position data over the SPI interface, providing each reading is taken shortly after activation of the sample indicator. If the timing of the host's SPI transactions is not synchronised to the sample indicator then the noise in reported position will be greater, due to additional delay compensation done in the Interface Processor. In this case the "DDC=0, encoder outputs" trace applies instead.

As noted in section 13.1, the Interface Processor adds its own delay compensation when generating encoder edges and when DDC=0. This adds position noise because of the additional delay compensation, and because the velocity estimate used for that compensation includes its own noise. The improvement in Noise Free Resolution is therefore less when delay compensation is applied, and the improvement is less the greater the delay that is compensated for.

The quality of the Motion Filter's velocity estimate increases with FILTLVL, and this means that the quality of delay compensation also improves. This means that there is less difference in Noise Free Resolution improvement between DDC=0 and DDC=1 for higher FILTLVL settings.

Where resolution is a customer's highest priority it is recommended to operate with delay compensation disabled, DDC=1, or with MINFILTLVL set to 150 or above.

13.3 Performance at Constant Velocity

Constant Velocity Time Delay is defined as the apparent time delay between the instantaneous position of the target and the position reported over the CAM622's encoder outputs ABN or the SPI interface, when the target is rotating at constant velocity.

By default, the CAM622 compensates for internal delays (DDC=0). Alternatively the CAM622's Motion Filter may be configured to disable Delay Compensation (DDC=1). This is typically done where resolution is an absolute priority or where the host applies its own compensation.

Table 65 Constant Velocity Delay (INTERVAL=300)

Setting	Interface	Constant Velocity Time Delay	Comments
DDC=0	ABN	0µs ±2µs	"Zero Phase Delay" with Delay Compensation active
	SPI, BiSS		Latch Point Compensation is applied, see section 13.6. INTERVAL must be set to 300.
DDC=1	ABN	58µs ±2µs	Delay Compensation disabled
	SPI, BiSS	20µs min, 50µs max	

Delay may also be expressed as an angle, and the relationship between the two are given by Equation 28:

Equation 28

$$Constant\ Velocity\ Angle\ Lag\ (°) = Constant\ Velocity\ Time\ Delay\ (s) \times Velocity\ (°/s)$$

Note that the velocity units in Equation 28 are degrees per second. Equation 28 may be rewritten with velocity in rpm, as in Equation 29:

Equation 29

$$Constant\ Velocity\ Angle\ Lag\ (°) = Constant\ Velocity\ Time\ Delay\ (s) \times Velocity\ (rpm) \times 6$$

13.4 Performance with Step Change in Velocity

Figure 44 illustrates how the Motion Filter responds to a step change in actual velocity, for DDC set to 0.

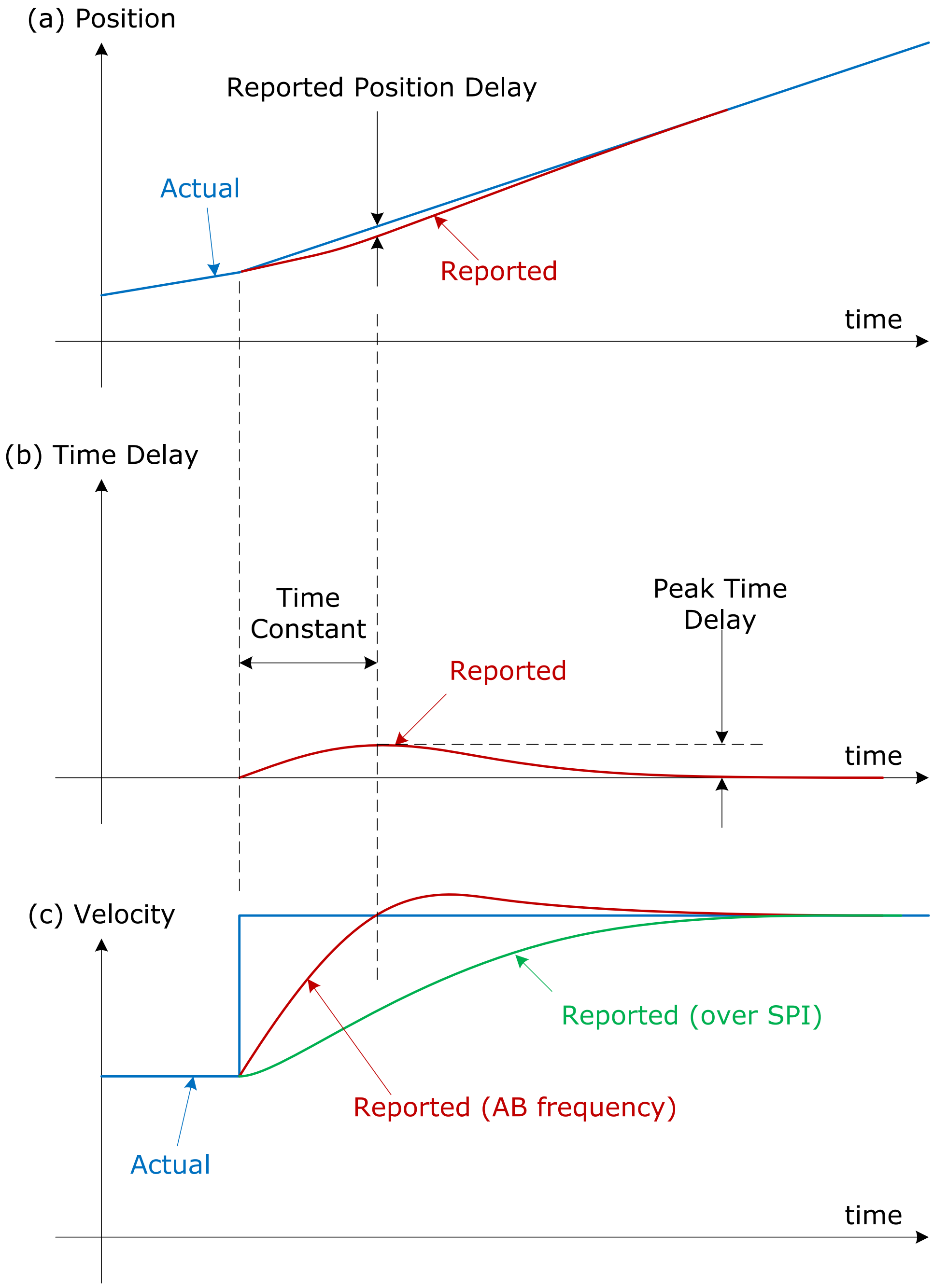


Figure 44 Response to step change in velocity

Figure 44(a) is a plot of position against time. The "Actual" trace is the target's position. The slope of trace increases where the velocity step change occurs. The "Reported" trace is the CAM622's position output. This represents the encoder output ABN count value (with discrete steps replaced with a continuous line for clarity), or the position read over SPI. Note how the Reported trace lags behind Actual for a short period, before catching up.

Figure 44(b) is a plot of the apparent time delay in reported position. It is equal to the difference between actual and reported position from Figure 44(a) divided by the instantaneous velocity. Both axes are in units of time, and they are scaled equally. The Reported Time Delay peaks at a value Peak Time Delay. This peak occurs at a time denoted Time Constant in Figure 44(b); it is equal to the Motion Filter's time constant.

Figure 44(c) is a plot of velocity against time. Actual Velocity steps upward abruptly. There are two "Reported" traces. "Reported (AB frequency)" is the instantaneous frequency of the CAM622's encoder AB pulses, converted to velocity units. Note that this frequency actually changes in discrete steps, every sample interval (nominally 30μs, see section 6.5). Steps are omitted for clarity. The "Reported (AB frequency)" trace also reflects the velocity a host would infer from successive position readings taken over SPI.

The "Reported (over SPI)" trace in Figure 44(c) is the velocity value read out over SPI, and for Latch Point Compensation (section 13.6). Note how this is slower to respond to the step change in velocity, as described in 13.1.

Figure 45 shows how the filter's Time Constant and Peak Time Delay vary with FILTLVL, with DDC set to 0.

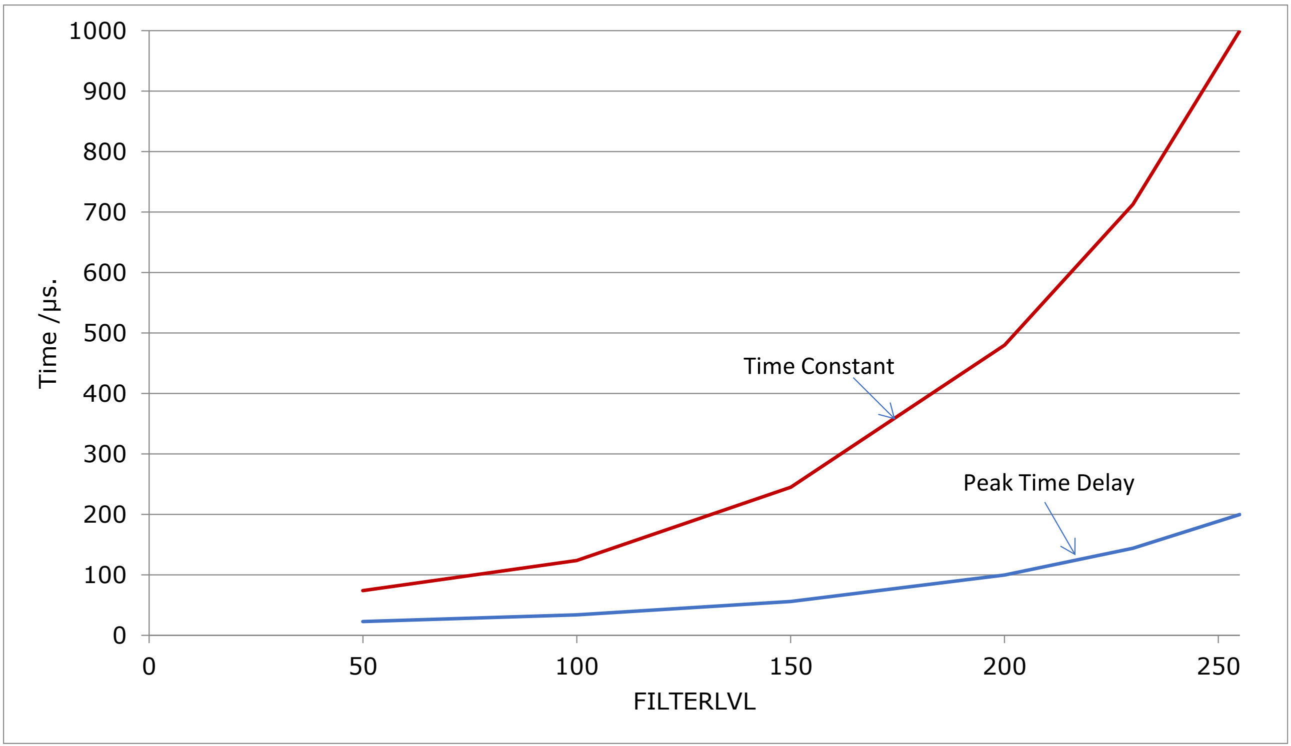


Figure 45 Time Constant and Peak Time Delay as a function of FILTLVL

13.5 Adaptive Filtering

Adaptive filtering varies the amount of filtering applied depending on detected motion. It aims to apply a high level of filtering when the target is stationary or moving at constant velocity, to maximise resolution and hence to minimise position noise. When acceleration is detected, adaptive filtering aims to reduce the amount of filtering, to reduce the filter's Time Constant and Peak Time Delay.

There are three controls for configuring adaptive filtering: MAXFILTLVL, MINFILTLVL and ADAPTSENS, as listed in Table 64.

When there is no detected acceleration, the Motion Filter uses a filter level FILTERLVL=MAXFILTLVL. MAXFILTLVL should be set to the minimum needed to achieve sufficiently low position noise output by the Motion Filter. In a system generating ABN encoder edges, the peak to peak position noise should usually be less than the angle between AB edges. This ensures that AB edges can be free from jitter due to position noise, when an appropriate amount of hysteresis is also applied, see section 9.7. Figure 43 may be used as a guide to how resolution improves (and peak to peak position noise reduces) with FILTLVL setting.

MINFILTLVL is the minimum value of filter level FILTERLVL that the Motion Filter will apply when it detects acceleration. To determine an appropriate value, first establish the maximum allowable filter Time Constant or Peak Time Delay for the Motion Filter, during periods of acceleration. Then select the corresponding value of FILTERLVL from Figure 45, and make this value MINFILTLVL.

If this value of MINFILTLVL is greater than or equal to the value of MAXFILTLVL established above then there is no need for adaptive filtering. Instead, use a value MINFILTLVL= MAXFILTLVL and set ADAPTSENS to 0. If this value of

MINFILTLVL is less than MAXFILTLVL then adaptive filtering is beneficial, and the next step is to establish an appropriate value for the ADAPTSENS parameter.

ADAPTSENS controls how sensitive the Motion Filter is to acceleration. It also controls how sensitive the Motion Filter is to position noise, since it has no way to distinguish position noise from acceleration. Position noise and acceleration both affect the Motion Filter's Pmeas input, and from the Motion Filter's perspective they are both random events that are impossible to anticipate. In practice, the choice of ADAPTSENS value should be made by considering position noise and not acceleration. This ensures that the Motion Filter does not adapt to position noise, and only to "genuine" acceleration.

A system's position noise may be measured using readings taken from a CAM622 over SPI with a stationary target. Filtering should be disabled (see section 13.7) so that the CAM622 reports raw measurement samples from position calculations done in the Sensing Engine ("Pmeas" in Figure 42). Please refer to section 7 for how to take measurements over SPI, and section 7.5 for how to interpret results including position. This is typically done during design. Once a worst-case value for peak to peak noise in Pmeas has been established, the corresponding value for ADAPTSENS may be taken from Figure 46.

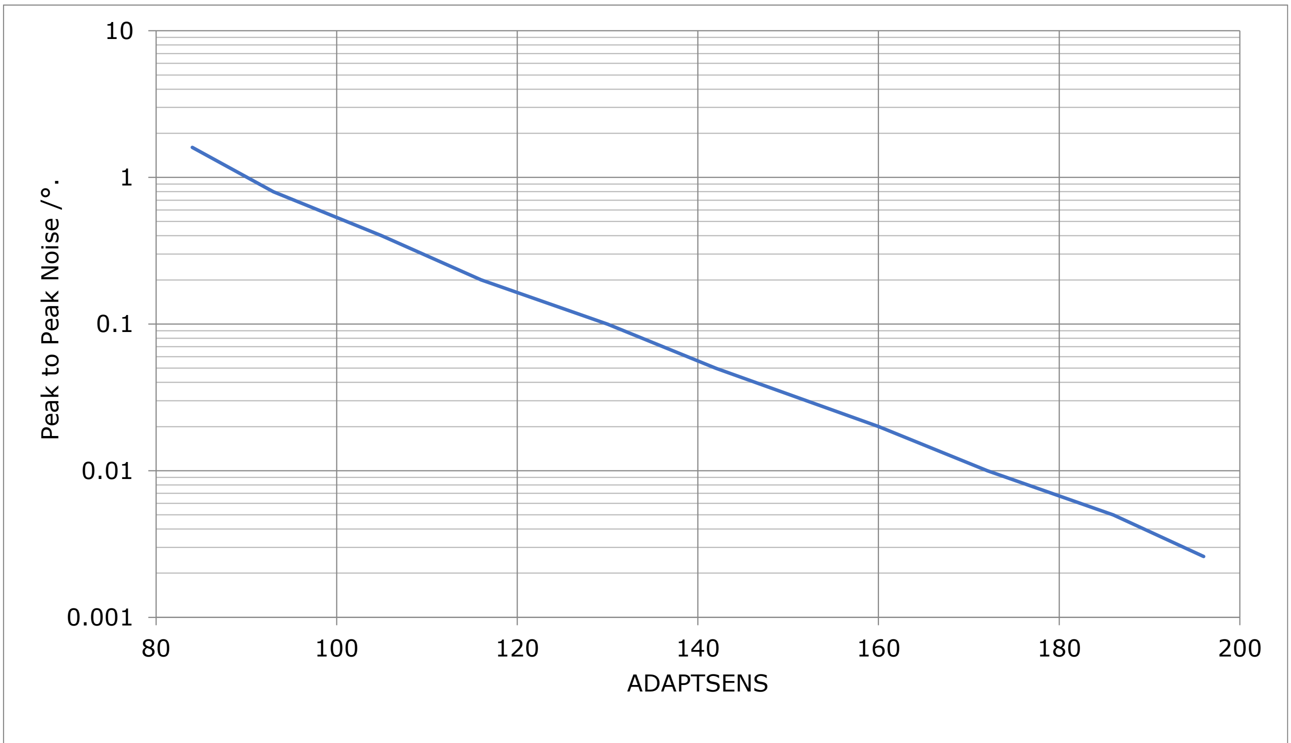


Figure 46 Maximum position noise as a function of ADAPTSENS to avoid unwanted adapting to noise

The effective amount of position noise may increase when the target is moving at high constant velocity due to sensor non-linearity. In this case systematic noise due to sensor non-linearity adds to random noise that is always present, including when stationary. This tends to reduce the Motion Filter's filter level FILTLVL when the target is moving at high velocity. In many applications this will be acceptable, because high resolution is rarely needed when operating at high constant velocity. If not, the value of ADAPTSENS may be reduced to accommodate the combination of random plus non-linearity induced noise, so that the Motion Filter does not adapt when the target is moving at high constant velocity.

The Motion Filter's FILTLVL value may be monitored by reading its value over the CAM622's SPI interface. Please refer to section 7 for how to take measurements over SPI, including readings from the FLTLVL register (Table 13). This may be done for diagnostic purposes, during the design process. When the target is stationary, FILTLVL should equal MAXFILTLVL. If smaller values are observed this may be a symptom that ADAPTSENS is too high. When the target's acceleration is sufficient the value of FILTLVL should decrease, down to a minimum of MINFILTLVL.

Figure 47 shows the result of a simulation illustrating how adaptive filtering helps reduce the apparent time delay of the Motion Filter. The upper red trace is without adaptive filtering. The lower blue trace is with adaptive filtering. In both cases there is a step change in velocity at time=0, and the graph plots apparent delay like in Figure 44(b). Adaptive filtering reduces the peak of apparent delay from 190µs to 40µs immediately following the change in velocity.

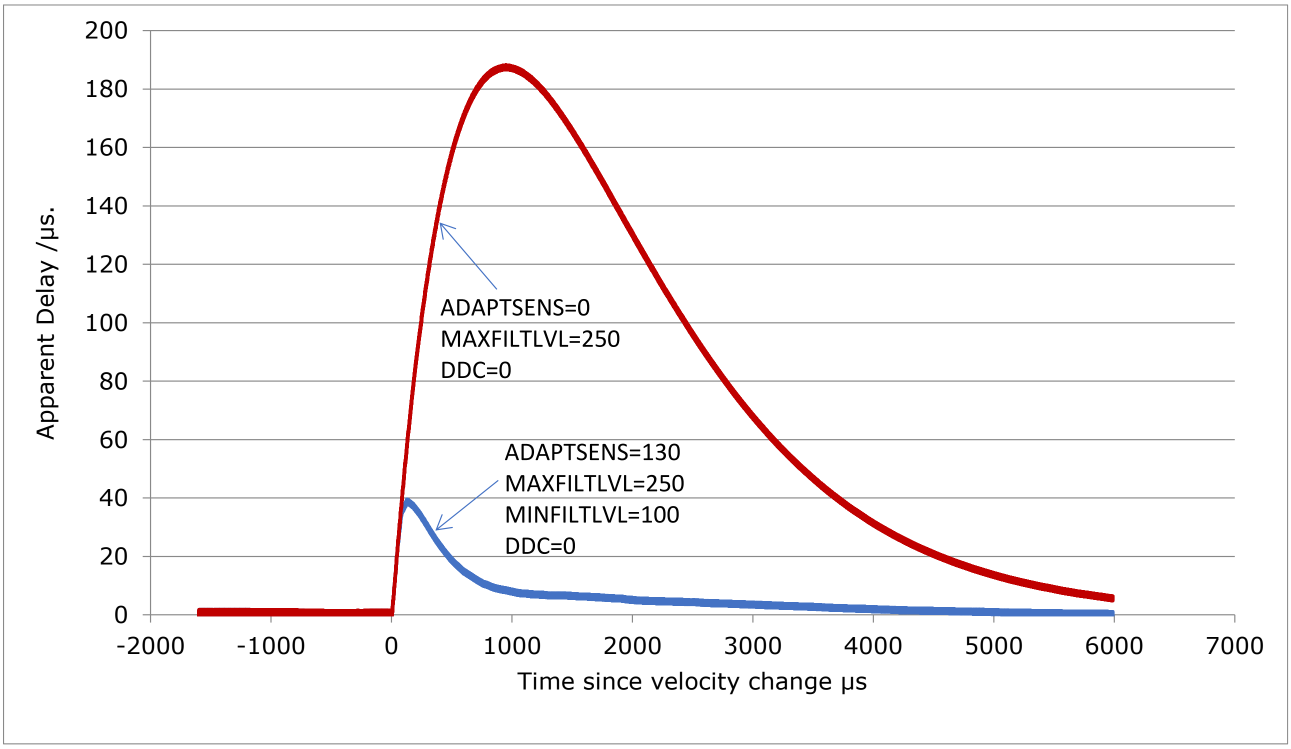


Figure 47 Effect of Adaptive Filtering

With no adaptive filtering, when ADAPTSENS=0, FILTLVL remains 250 throughout.

With adaptive filtering, when ADAPTSENS=130, FILTLVL is 250 before the velocity change (=MAXFILTLVL). FILTLVL reduces to 100 during the velocity change (=MINFILTLVL). It then ramps upwards back towards 250, so that it is back to 250 (=MAXFILTLVL) 3000µs after the velocity change. These changes serve to reduce the peak apparent delay from 190µs to 40µs. They also lead to a large reduction in the time taken to react to the velocity change.

Figure 47 illustrates the benefits of Adaptive Filtering when there is an instantaneous change of velocity, and therefore represents a system with very high acceleration and dynamic behaviour. Adaptive filtering will usually only be of benefit when accelerations are high. For example greater than 100,000rad.s^{-2} when MAXFILTLVL is 250, or 1,000,000rad/s^{2} when MAXFILTLVL is 170. When acceleration is lower, the apparent time delay is much less because the motion filter can keep up with the velocity change easier. It is then simpler to select a fixed MAXFILTLVL value.

13.6 Latch Point Compensation

The CAM622 takes discrete position measurements, Sample Interval apart. As illustrated in Figure 48, measurement results are delayed relative to Current Position by Processing Time.

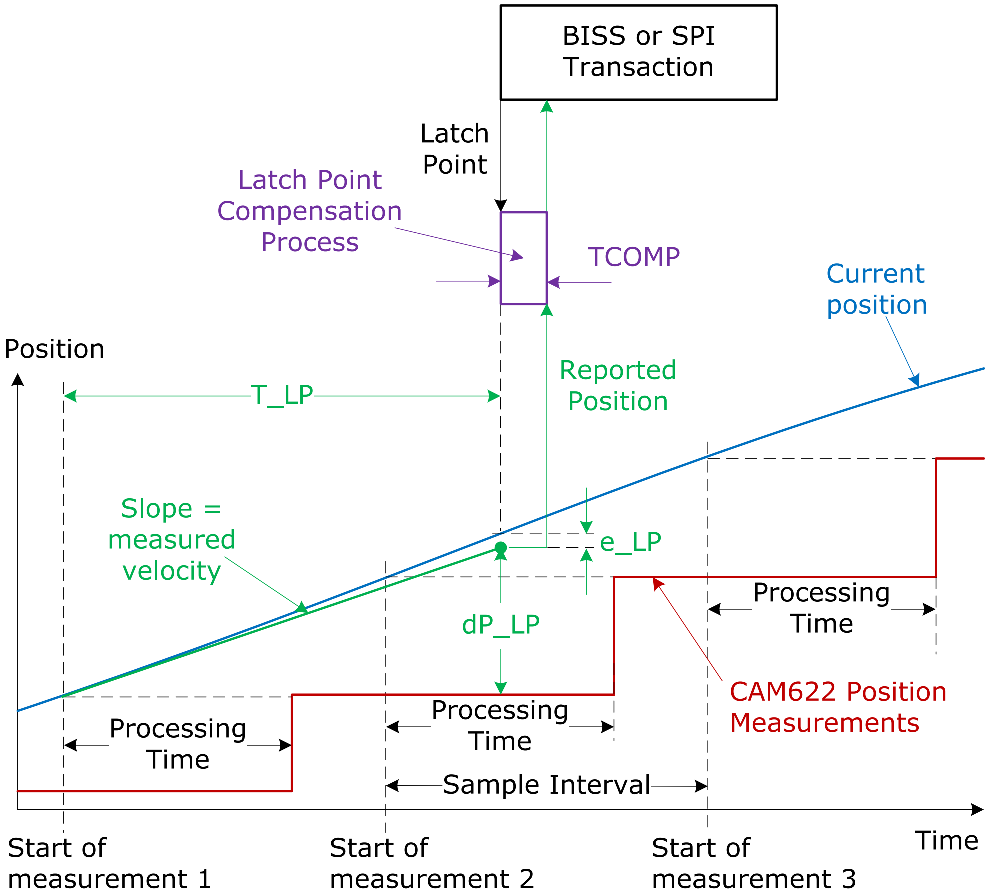


Figure 48 Latch Point Compensation Process Overview

When the Disable Delay Compensation bit is set to 1, the CAM622 reports the most recent position measurement when it is requested over the SPI or BiSS interface. In this case the reported position is subject to a variable delay T_LP specified in Table 65. The variability is caused by the asynchronous nature of the SPI/BiSS transaction and the CAM622 measurement process. This is acceptable in many applications, especially where velocity is low. However in a highly dynamic system operating at high velocity the delays and resulting apparent position jitter may be unacceptable.

When the Disable Delay Compensation bit is set to 0, the CAM622 performs Latch Point Compensation. It measures the time T_LP between the most recent measurement result's capture of position values and the SPI/BiSS transaction's Latch Point. The Latch Point is defined as the negative going edge of nCS for an SPI transaction, and the BiSS Latch Point illustrated in Figure 35 for BiSS. The CAM622 then multiplies its velocity measurement by T_LP to yield a position correction dP_LP. It then adds this correction to the most recent measured position value and reports it over SPI/BiSS.

The CAM622 does not attempt to compensate for acceleration, so a residual error illustrated as e_LP in Figure 48 will be present when accelerating. However since the Filter Time Constant, Sample Interval and Processing time are short, it takes extremely high accelerations for this error to become significant. Adaptive Filtering (section 13.5) is recommended for systems with high acceleration, since it detects acceleration and reduces the filter level and hence Filter Time Constant accordingly. Adaptive Filtering helps maintain Noise Free Resolution when acceleration is not present. Where a system's dynamic performance is more important than Noise Free Resolution, MAXFILTLVL can instead be set to a small value. Avoid values less than 50, otherwise velocity noise may become excessive, harming the quality of Latch Point Compensation.

The time taken to perform Latch Point Compensation TCOMP is on the order of 1.5μs, which is fast enough to guarantee that the corrected position value will be available to clock out in time over both SPI and BiSS.

In the case of SPI, Latch Point Compensation will be complete by the time the host has read out the first word over SPI containing the CTRL register (section 6.3). By the time it starts clocking out the second word containing the top 16 bits of position data Latch Point Compensation will be complete, even with the fastest SPI settings allowed in Table 11.

In the case of BiSS, the iC-MCB device delays transmission of the START bit until it is ready to transmit position data. As illustrated in Figure 35 the Latch Point Compensation Process is only one contribution to this delay, with data transmission from CAM622 to iC-MCB and the iC-MCB's own internal delays dominating.

Figure 48 illustrates a single SPI/BiSS transaction in the space of 3 measurements. In fact, the CAM622 can perform Latch Point Compensation multiple times during a single measurement, so that the CAM622 delivers position data to the host device faster than it takes measurements.

The calculation done by the Latch Point Compensation process assumes that INTERVAL is set to 300 to yield a Sample Interval of 30µs. This means that INTERVAL must be set to 300 to yield correctly compensated position values.

13.7 Disabling the Motion Filter

The motion filter may be disabled with the settings shown in Table 66. These settings make the motion filter's output (Pout in Figure 42) equal to its input (Pmeas). This allows a host device to take charge of any filtering and/or delay compensation that is required.

These settings are also required when taking measurements of raw measurement noise. That is, the noise in the Position Calculation output as shown in Figure 4. This is the basis for measurements of noise free resolution quoted in sensor datasheets.

Table 66 Motion Filter settings to disable filter

Setting	Value
MAXFILTLVL	0
MINFILTLVL	0
ADAPTSENS	0
DDC	1

14 Performance

14.1 Supply Current

Table 67 provides guidance on the supply current drawn by the CAM622 for various operating conditions. Figures include supply current drawn by the CAM622 itself and the excitation circuitry. That is, the current passing through both R_S and R_EX of Figure 6.

Table 67 Supply current

Sample Interval	Operating condition	Typical average supply current
30µs	Generating ABN signals autonomously	115mA
30µs	Generating SENT signals autonomously	110mA

14.2 Thermal Stability

Sensors, targets and the CAM622 IC are all individually stable across temperature. None have any form of temperature compensation, and temperature stability arises purely by the symmetry of design of each element.

System-level characterisation was carried out on the 25mm B3 rotary sensor and target together with CAM622 circuitry. Please see the 25mm B3 precision rotary sensor datasheet for details. In summary, reported position changed by less than 0.01° across the entire temperature range -40°C to +125°C.

15 Bootloader Operation

The CAM622 chip has embedded software inside. This is partitioned into two fields: Application Code and Bootloader Code. Application Code is responsible for normal operation including taking measurements and interfacing the results to a host device. The Bootloader Code allows a host device to update the Application Code using the chip's SPI interface. In normal operation, the version number of the Application Code (the System Version Number) can be read over the SPI interface from the SYSVER register. The version number of the Bootloader Code can be read from the BOOTVER register. The version number of the Sensing Engine can be read from the SEVER register. Please see Table 14.

Please refer to document "Updating Application Code" for details of how to program Application Code using the bootloader. CAM622 timings specific to the bootloader and described in that document are specified Table 68.

Table 68 Bootloader timings

Parameter	Description	Min	Max
TnRST2nCS	Delay between end of reset and first SCK high in order to enter bootloader mode.	3.2ms	
TBOOTPIN	Pause required between 2nd and 3rd Data Block words	500µs	
TSDOL2SCKH_BOOT	Time between each Data Block Word and the next	5.0 µs	
TBOOTWAIT	Variable pause required after each data block	20ns	20ms
TBOOTLOAD	Overall time to update Application Code	1.4s	

The SPI parameters specified in Table 11 are the same for Bootloader SPI as for normal operation.

Note that the maximum specified value of TBOOTWAIT above only occurs a few times in the upload process. TBOOTWAIT is more typically at or near its minimum value.

The total time taken to undertake the complete Bootloader process is denoted TBOOTLOAD. Table 68 specifies a minimum value, based on a host responding to SDO low immediately and operating with the minimum TSDOOL2SCKH_BOOT value. TBOOTLOAD(min) is the time the bootload process takes for a sufficiently fast and well optimised host.

16 Package Details

16.1 CAM622UE

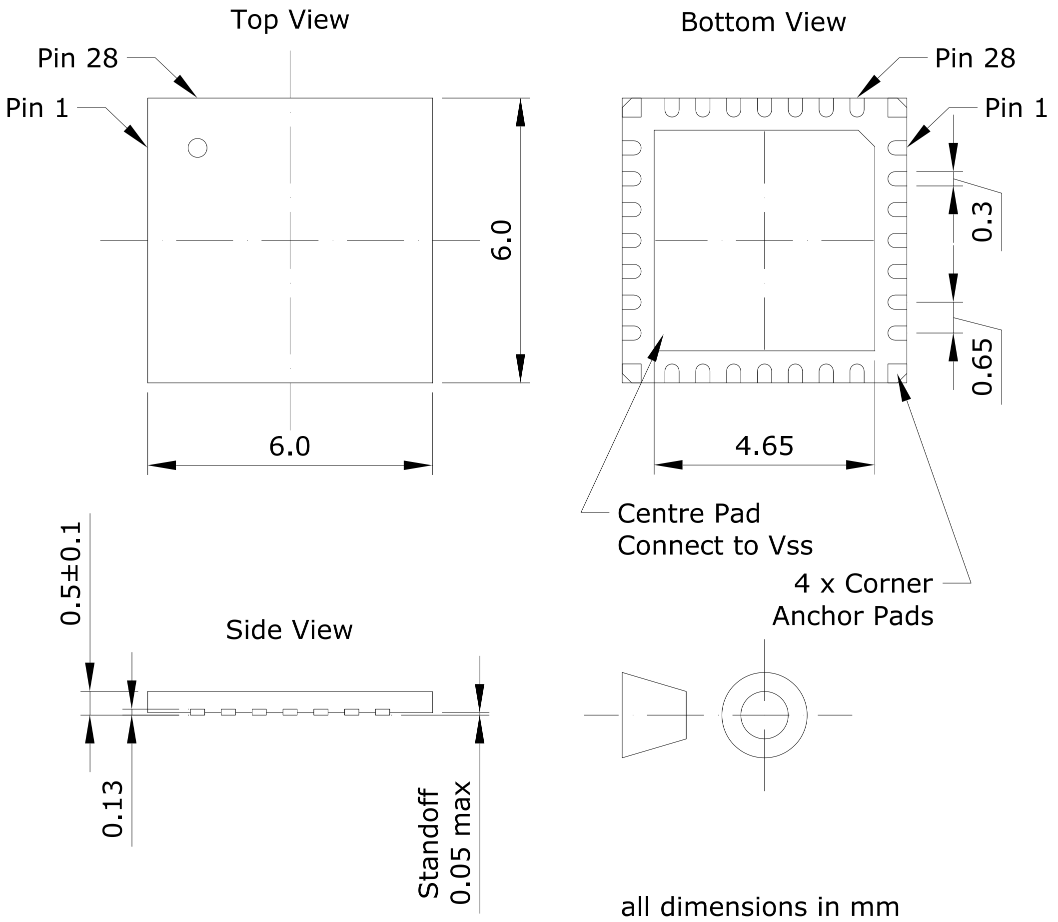


Figure 49 CAM622UE plastic quad-flat no-lead 28-pin (UQFN: U suffix)

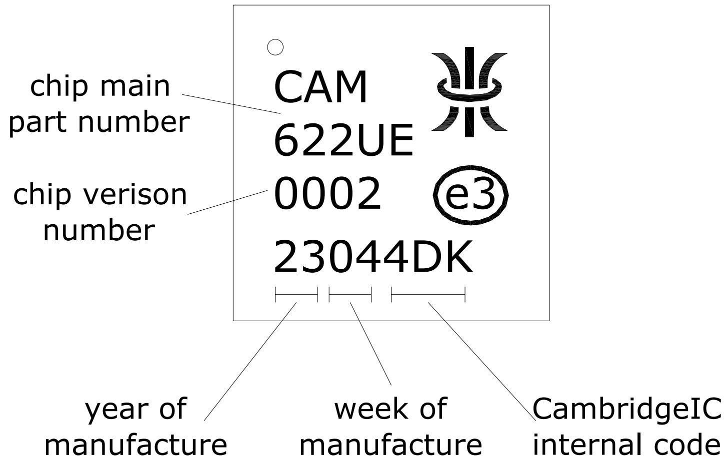


Figure 50 CAM622UE product markings

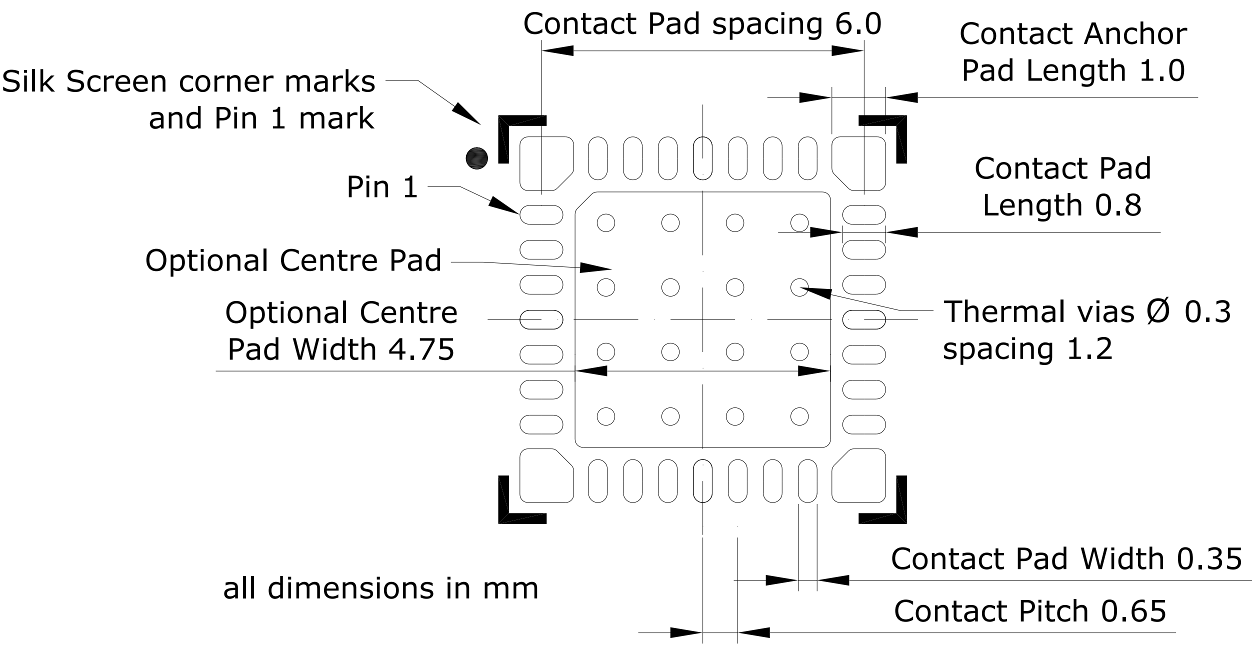


Figure 51 CAM622UE Recommended PCB Footprint

The CAM622UE has 4 Corner Anchor Pads and a Centre Pad. Connect the Centre Pad to VSS. The Corner Anchor Pads are internally connected to the Centre Pad. It is recommended to leave them unconnected on the PCB.

17 Tape and Reel Specifications

17.1 CAM622UE

CAM622UE chips are available in tape and reel on complete reels of 3300 parts. The carrier tape is illustrated in Figure 52, and dimensions are specified in Table 69.

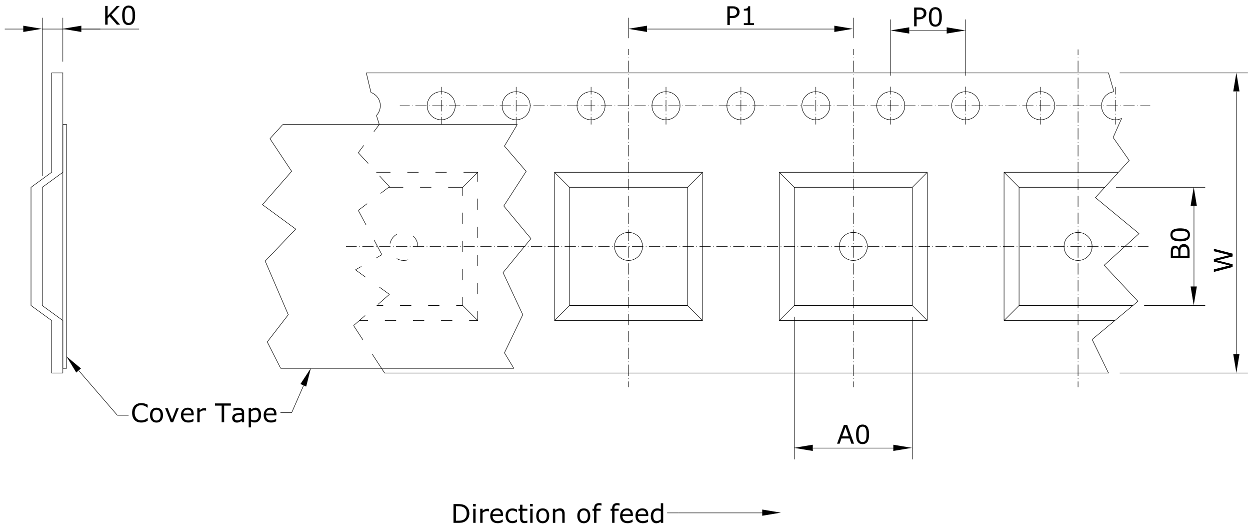


Figure 52 CAM312ME carrier tape dimensions

Table 69

Tape and Reel Specifications		**Dimensions in mm**						
Package	**Units per reel**	**Reel diameter**	**W**	**P0**	**P1**	**A0**	**B0**	**K0**
28-pin UQFN	1600	330	16	4	12	6.3	6.3	0.80

18 Reflow Soldering Recommendations

The CAM622 is available in lead free packaging only. The recommended reflow soldering temperature profile is illustrated in Figure 53. Values are shown in Table 70.

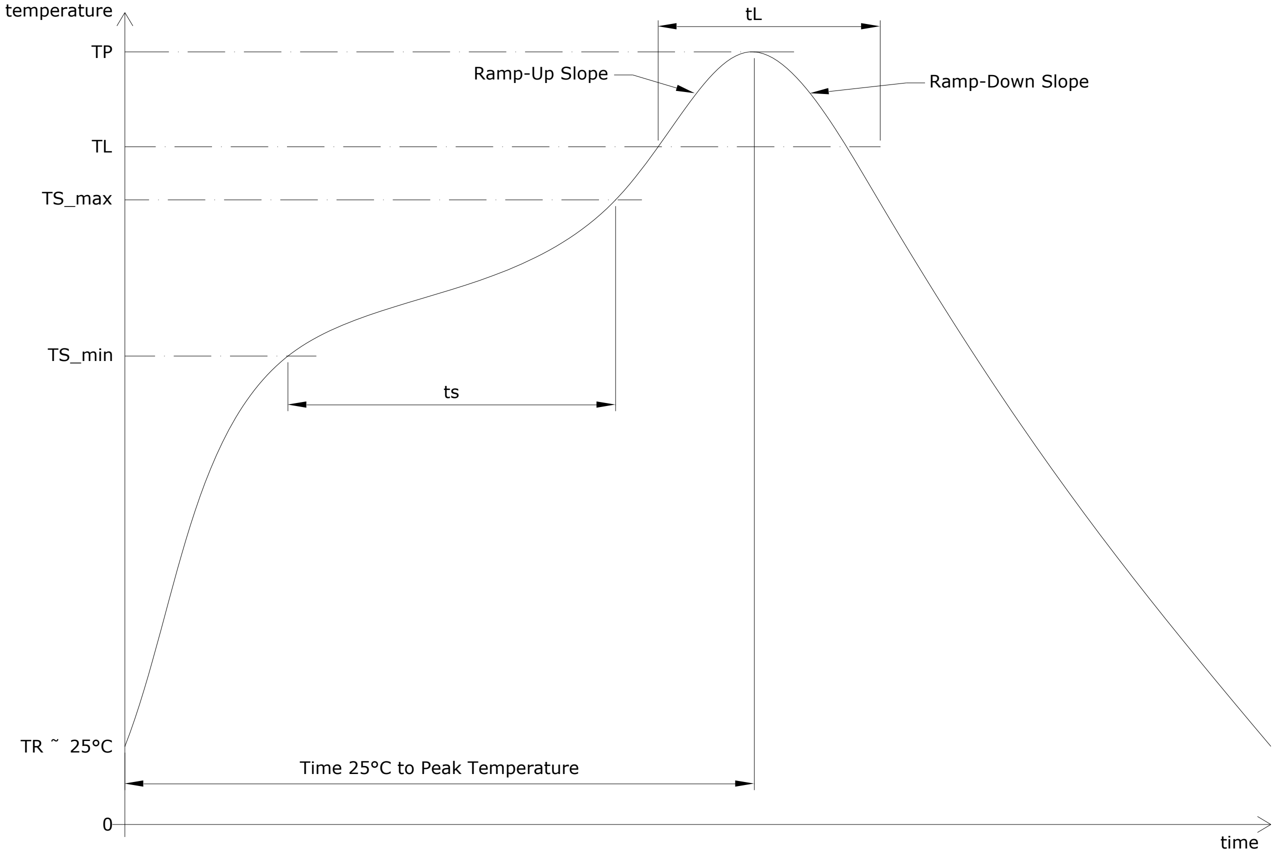


Figure 53 reflow soldering temperature profile definitions

Table 70

Profile feature	**Value**		**Comments**
TS_min	150°C		Preheat temperature range
TS_max	200°C		
ts	60s min	120s max	Preheat time
TL	217°C		Liquidous temperature
TP	225°C min	260°C max	Peak temperature
tL	60s min	150s max	Time maintained above Liquidous temperature
Ramp-Up Slope	3°C/s max		
Ramp-Down Slope	6°C/s max		

19 Environmental

Table 71

Item	**Min**	**Max**
Storage temperature	-65°C	160°C

20 RoHS Compliance

The CAM622 uses Matte Tin (Sn) pin finish. CambridgeIC certifies, to the best of its knowledge and understanding, that the CAM622 chip is in compliance with EU RoHS directive 2011/65/EU and 2015/863.

21 Document History

Table 72 main changes

Rev	**Date**	**Comments**
0005	1 Aug 2025	Added interpretation of linear position and velocity for a linear sensor Changed Q1, Q2 MOSFETs from obsolete FDT4000CZ to Si1016CX Updated Sensor coil and filter connections drawing to reposition C_B correctly
0006	8 Dec 2025	Increased number of parts on complete reel to 3300 Moved external circuitry descriptions to the sections they relate to The following additions apply only to Application Code V2.0 onward: Added BiSS interface as an option Added Latch Point Correction Corrected location of INDEX3 and INDEX2 in INDEX32 register Added SENT interface details Increased TnCSH_read(min) for when ABEN=0 from 2.5μs to 4.0μs
0007	15 Jan 2026	Minor corrections Renamed BISSKEY2 to BISSKEY0 for consistency with other register names
0008	28 Jan 2026	Introduced alternative TnCSH_START value following clearing of START bit with ABEN=1 Specified a minimum SENTTICK value Increased Minimum BiSS Cycle Time and t_busy Added "The combination SENTOD=1 and SENTPOL=1 is not allowed"

22 Contact Information

Cambridge Integrated Circuits Ltd
21 Sedley Taylor Road
Cambridge
CB2 8PW
UK

Tel: +44 (0) 1223 413500
info@cambridgeic.com

23 Legal