Description

CambridgeIC's CAM312 Central Tracking Unit (CTU) chip is a single-chip processor for position measurement. It implements the electronic processing for resonant inductive position sensing technology.

The CAM312 measures the position of contactless, inductively coupled targets relative to sensors that are built from printed circuit boards to CambridgeIC's designs.

The CAM312 supports a wide variety of different sensors, including rotary, linear and arc.

Features

- Resonant inductive position sensing engine
- Fully ratiometric measurements
- Automatic tuning to target frequency
- Fast SPI communications (slave device)
- User IOs for sample indicators
- Measures two Type 4 sensors or one Type 2, 3 or 6
- SPI interface checksum available for integrity check
- Analog outputs possible using an external DAC
- Power down mode for reduced supply current
- Internal software upgradable over SPI

Performance

- Up to 2000 position samples per second
- Resonator frequency tuning range at least ±10%
- Short Group Delay of 200µs
- Noise Free Resolution 10…15 bits depending on sensor

Product identification	
Part no.	**Description**
CAM312ME	28-pin QFN -40°C to 125°C
CAM312MA	28-pin QFN -40°C to 150°C

Figure 1 CAM312ME image

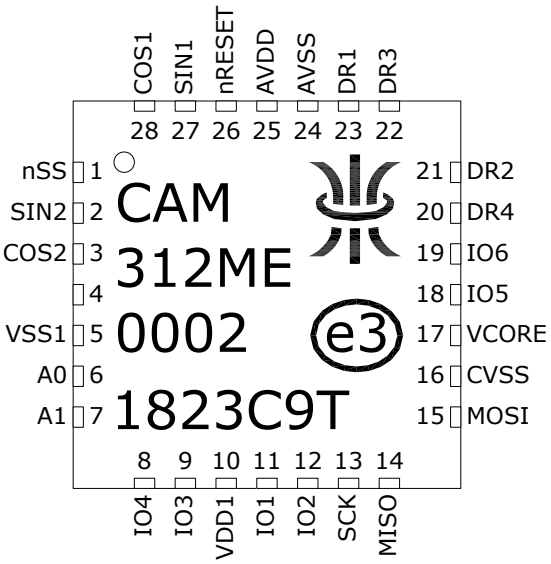


Figure 2 CAM312ME 28-pin SSOP pinout

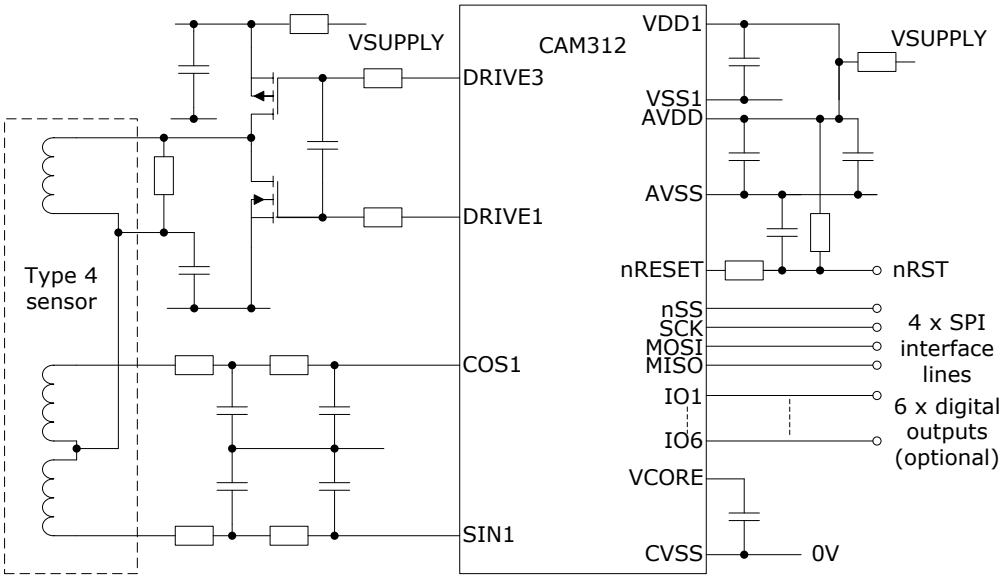


Figure 3 Circuit for reading a single Type 4 sensor

1 CAM312 Functional Description

1.1 Overview

The CAM312 Central Tracking Unit (CTU) chip works with a sensor built from a PCB to measure the position of contactless targets. Targets comprise an inductively coupled resonant circuit. Sensors are available for linear, rotary and arc measurement and in a range of different sizes.

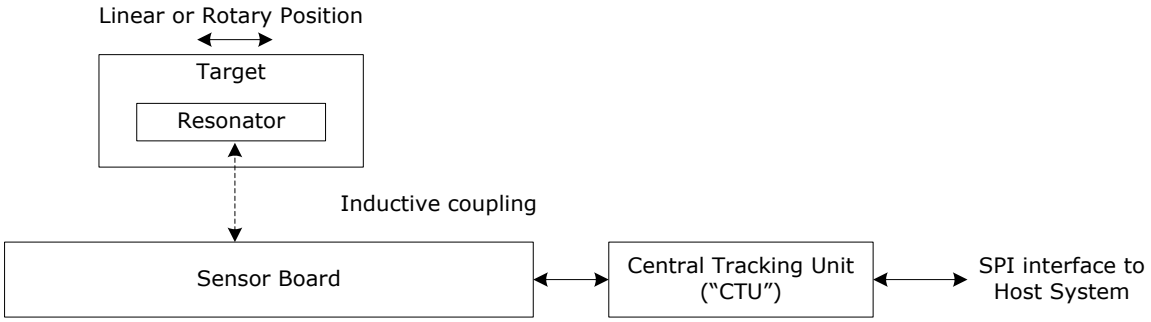


Figure 4 Resonant inductive position sensing system

Sensor Types differ in their details, but the same basic measuring principle applies to them all. The CAM312 chip generates signals that drive external miniature MOSFETs. These, in turn, drive AC current into the sensor's excitation coil. The excitation coil current generates an AC field which powers the resonator at its resonant frequency. The energy in the resonator is built up during this pulse.

Then the current is removed, and the resonator induces decaying EMFs in the sensor coils. The CAM312 chip detects the amplitude of this echo in each sensor coil. It then uses the amplitude values to calculate position.

The details of the measuring process and calculation depend on sensor Type. In all cases, sensing and calculation are fully ratiometric for immunity to changes in amplitude due to gap, misalignment, temperature, target frequency and supply voltage.

The pulse echo interrogation method separates the excitation and detection processes in time. This yields immunity from stray coupling between excitation and sensor coils, and superior sensing performance.

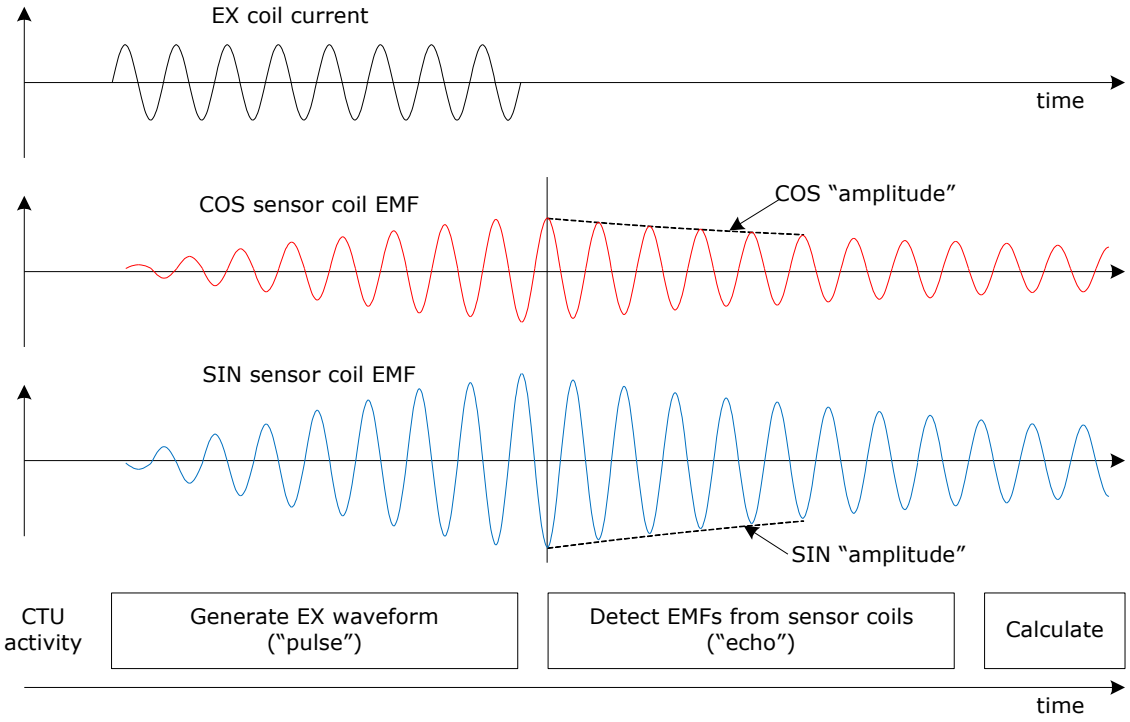


Figure 5 Electronic measuring process

1.2 Sensor Types

Table 1 lists the sensor types currently supported by the CAM312, and includes section references for more details on each Type.

Table 1 Comparison of sensor types

Type	Number of coils in each sensor		Applications	Max number of sensors (CAM312)	Has Subtypes?	Section number for details
	Excitation	Sensing				
2	1	4	Long precision linear sensors	1	Yes	5
3	Total of 4 coils		Linear sensors with free rotating target e.g. for float level	1	Yes	6
4	1	2	Simple linear, rotary and arc sensors	2	No	3
6	1	4	Precision linear and rotary sensors	1	Yes	4

1.3 Pin Names and Functions

Table 2 summarises CAM312 pin functions and type. Please refer to section 2 for electrical characteristics for each type.

Table 2 CAM312 pin names and functions

Signal Name	Type	Function
VDD	Power	Positive supply voltage.
AVDD	Analog Input	Analog supply voltage, connect to VDD.
VSS, AVSS, CVSS	Power	0V connection and common return for sensor inputs
VCORE	Power	On-chip 1.8V regulator decoupling capacitor connection
nRESET	Digital Input	Hardware reset, active low
nSS	Digital Input	SPI Interface line: Slave Select, active low.
SCK	Digital Input	SPI Interface line: Serial Clock.
MOSI	Digital Input, 5 V Tolerant	SPI Interface line: Master Out, Slave In.
MISO	Digital Output (1)	SPI Interface line: Master In, Slave Out.
IO1, IO2, IO4, IO5, IO6	Digital or Open Drain Output, 5V Tolerant	User configurable IO.
IO3	Digital or Open Drain Output	
DRIVE1 – DRIVE4	Digital Output	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		Do not connect
Pin 4	Not used	Do not connect
Exposed Pad	Shield	Large pad under package. Connect to Vss.

Note (1): MISO is driven as a digital output by the CAM312 during SPI transactions, when is nSS low. When nSS is high (inactive) it is Open Drain to allow other slave devices to share the SPI bus.

2 Electrical Characteristics

2.1 Operating Characteristics

Table 3 Operating characteristics

Item	Min	Max	Comments
Operating Supply Voltage VDD, AVDD	3.0V	3.60V	
Operating Temperature	-40˚C	125˚C	Ambient temperature, CAM312ME
	-40˚C	150˚C	Ambient temperature, CAM312MA
VDD start voltage relative to VSS		0V	For reliable power on reset
VDD rise rate relative to VSS	0.05V/ms		

2.2 Absolute Maximum Ratings

Table 4 Absolute maximum ratings

Item	Max
Voltage between VDD or AVDD and VSS	-0.3V to +4.0V
Voltage on 5V Tolerant pins relative to VSS when VDD≥3.0V	-0.3V to +5.5V
Voltage on 5V Tolerant pins relative to VSS when VDD<3.0V	-0.3V to +3.6V
Voltage on any other pin relative to VSS	-0.3V to (VDD+0.3V)
Current into or out of Digital Output (operating temperature ≤125°C)	4mA
Current into or out of Digital Output (CAM312MA, -40°C to 150°C)	2.5mA

2.3 Digital Input Specifications

Table 5 Digital input specifications

Item	Min	Max
Input Low	VSS	0.2 x VDD
Input High, 5V Tolerant	0.8 x VDD	5.5V
Input High, NOT 5V Tolerant	0.8 x VDD	VDD
Input leakage current		±2µA

2.4 Digital Output Specifications

Table 6 Digital output specifications

Item	Min	Max	Comments
Output Low Voltage		0.4V	IOL = 4mA
Output High Voltage (Digital setting)	2.4V		VDD=3.3V IOH = -4mA
Output High Current (Open Drain setting)		±2µA	
Output High Voltage, Open Drain, 5V Tolerant		5.5V	Maximum allowed
Output High Voltage, Open Drain		VDD	Maximum allowed

2.5 Application Memory Characteristics

Table 7 Application Memory characteristics

Item	Min	Max	Comments
Number of FLASH updates		2000	Across Operating Supply Voltage and Operating Temperature
Characteristic retention, -40°C to +125°C	20 years		

2.6 Hardware Reset

The CAM312 chip will be reset when the host device pulls nRST low, as illustrated in Figure 6. The timing parameter TnRSTL is the low time for the nRST signal. The CAM312 starts internal validity checks on the rising edge of nRST (section 11.1).

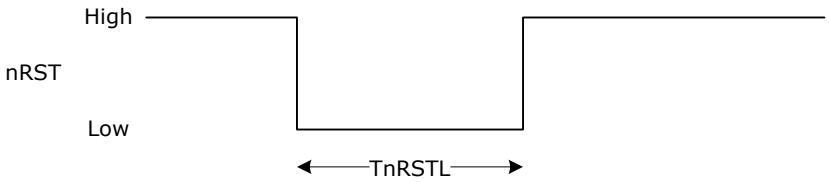


Figure 6 nRST low duration for reliable reset

Table 8 specifies the timing parameter nRSTL required for reliable reset.

Table 8 Reset timing

Parameter	Description	Min	Max	Units
TnRSTL	Duration of nRESET pin low to reliably reset IC	2.5		µs

3 Type 4 (and Type 1) Sensor Application

Type 4 sensors include an excitation coil to energise the resonator inside a target, and a single pair of COS/SIN sensor coils to detect the position of the resonator.

This circuitry is also for connecting Type 1 sensors. The only difference is that Type 4 sensors have a common VREF connection for all coils.

3.1 Circuit Schematics, Single Type 4

The circuit for connecting a single Type 4 sensor to the CAM312 chip is shown Figure 7 below. Component values are listed in Table 9.

VCORE is an external connection to an on-chip 1.8V regulator. This pin requires an external decoupling capacitor C_CORE. It must not be connected to other circuitry.

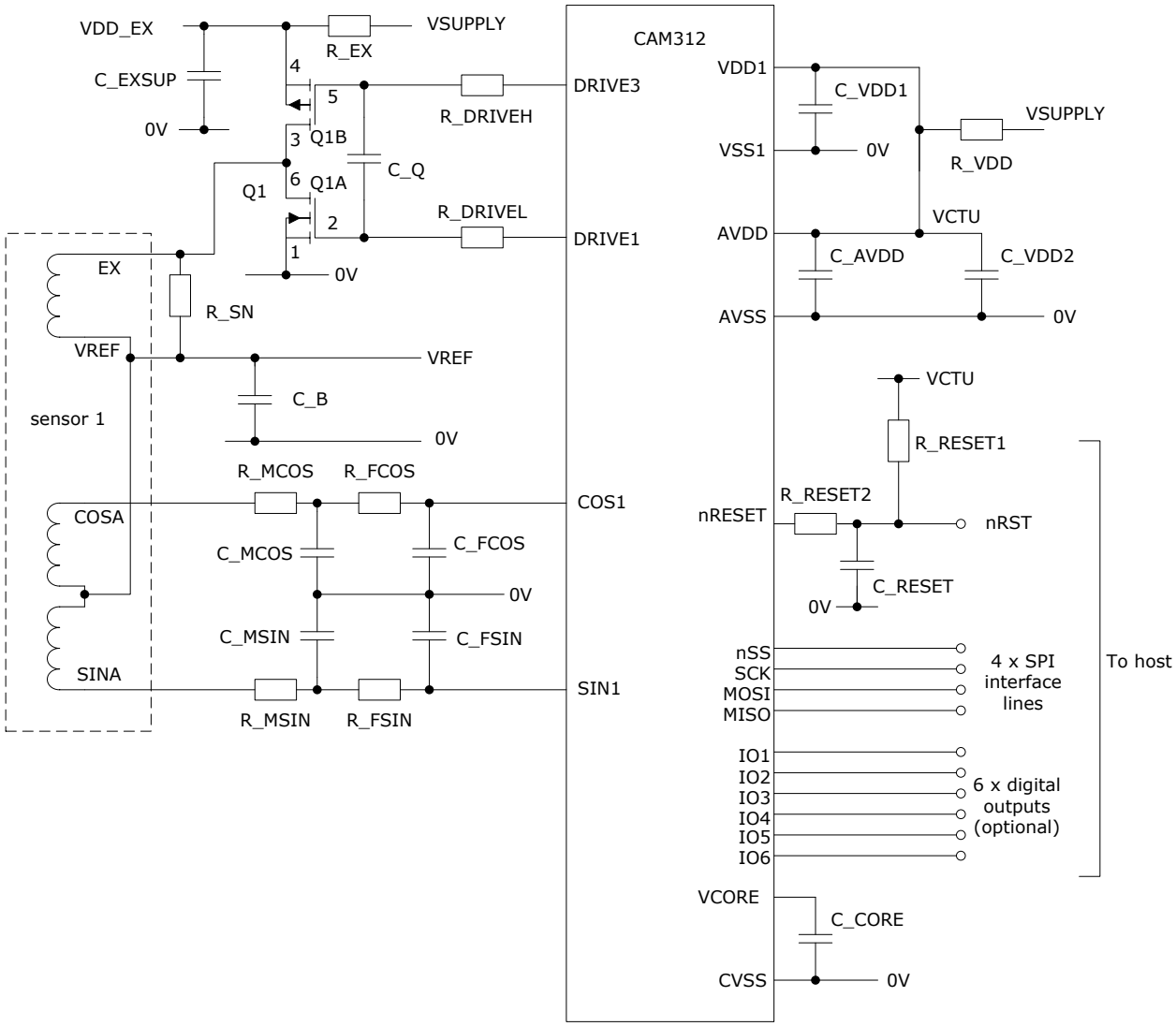


Figure 7 CAM312 connections, single Type 4 sensor

The CAM312's nRESET pin requires a series resistor R_RESET2 for current limiting in the event of ESD, a pull-up resistor R_RESET1 and a reset timing capacitor C_RESET. The host may also perform a reset, by pulling the signal nRST low with an open drain output. Connecting the reset line in this way simplifies bootloader operation, because this allows the host to use the "Reset Entry Method" when the CAM312 does not contain valid Application Code (section 14.4).

MISO is an open drain output when nSS is high, and requires a pull-up resistor of 4.7kΩ if this is not provided by the host.

The CAM312 uses an external MOSFET half bridge to drive a Type 4 sensor's excitation coil. MOSFET pair Q1A and Q1B is available as a single miniature device. The gate drive circuit uses 2 resistors and 1 capacitor, and is designed to enable the CTU to drive the bridge output to a high impedance state. The resistors R_DRIVE limit the operating speed of the MOSFETs, to minimise capacitively coupled emissions. The capacitor C_Q prevents excessive shoot-through current during switching. R_SN absorbs the energy in the excitation coil on the transition from low to high impedance. The energy required for each pulse of excitation current is stored in C_EXSUP. R_EX limits the peak charging current.

The sensor's COS and SIN coil inputs have two stages of RC filtering for immunity to external high frequency interference. The reference voltage VREF is generated by the switching action of the MOSFET driver. VREF should not be generated by other means.

3.2 Circuit Schematics, Additional Type 4

The CAM312 chip can process two Type 4 sensors. In this case the circuitry shown in Figure 8 is required in addition to that of Figure 7.

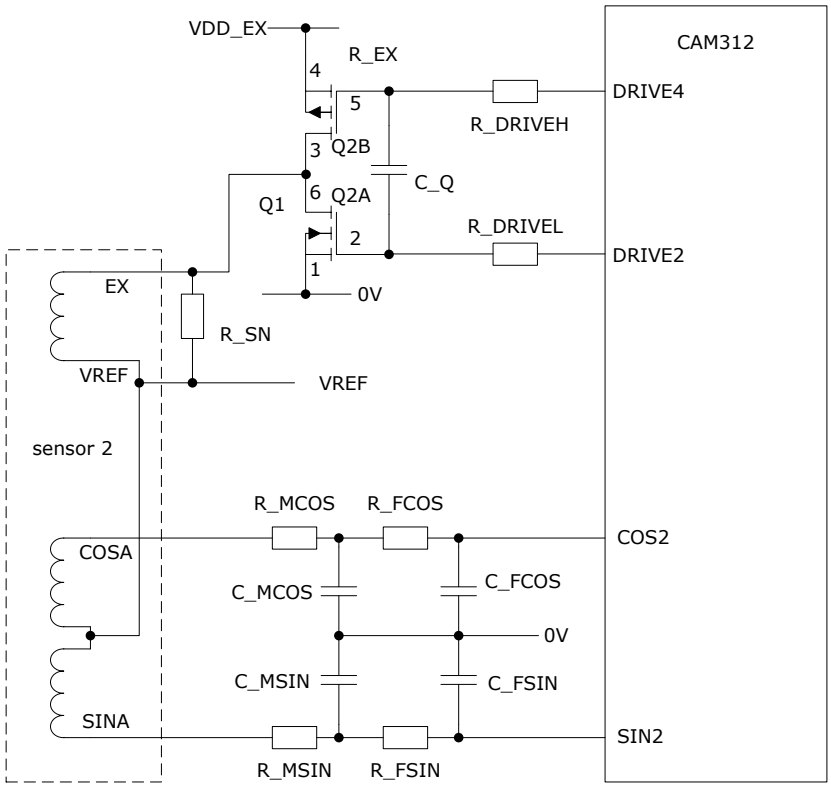


Figure 8 CAM312 connections, additional Type 4 sensor

The second sensor's excitation circuitry shares excitation decoupling capacitor C_EXSUP with the first. It also shares the same capacitor C_B used to establish the bias voltage VREF.

3.3 Components Required, Type 4

Table 9 lists the component values and numbers required for the schematics of Figure 7 and Figure 8.

Table 9 components required for Type 4 sensor connection

Circuit Ref	Value	Tolerance		Number required	
		Grade A	Grade B	1 sensor	2 sensors
R_RESET1	10kΩ	±5%		1	1
R_RESET2	470 Ω	±5%		1	1
R_VDD	1Ω	±5%		1	1
R_EX	4.7Ω	±5%		1	1
R_DRIVEL	1kΩ	±5%		1	2
R_DRIVEH	330Ω	±5%		1	2
R_SN	1kΩ	±5%		1	2
R_MCOS/SIN	150Ω	±0.1%	±1%	2	4
R_FCOS/SIN	150Ω	±0.1%	±1%	2	4
C_VDD1, C_AVDD	100nF	±10%		2	2
C_CORE, C_EXSUP, C_VDD2	10µF, ESR < 1 Ω	±20%		3	3
C_RESET	1nF	±10%		1	1
C_Q	1nF	±10%		1	2
C_MCOS/SIN	1nF	±1%	±5%	2	4
C_FCOS/SIN	1nF	±1%	±5%	2	4
C_B	4.7µF	±10%		1	1
Q1 (and Q2 if 2 sensors)	Si1016CX			1	2

The value of C_EXSUP may be increased in order to reduce peak supply current, see section 12.2.

The filter components connected to the CAM312 chip's sensor inputs are important for reproducibility. The table above includes two grades for them: A and B. Grade A yields the least system to system reproducibility error due to component differences. Grade B components are slightly more cost effective. Grade A and B reproducibility error is compared in Table 10.

Table 10 reproducibility error due to filter components, Type 4

Grade	Reproducibility due to filter components (% of SinLength)
A	±0.04%
B	±0.2%

The SinLength value depends on the sensor, and is quoted in its datasheet. For 360° Type 1 or 4 rotary sensors it is usually 360°. For linear sensors, it is usually a few mm more than the measuring range of the sensor.

4 Type 6 Sensor Application

Type 6 sensors are for the precise measurement of rotary and linear position. Their additional precision comes from fine ("A") sensor coils whose signals repeat multiple times along the measuring path. This multiplies resolution and accuracy. Coarse ("B") sensor coils provide full absolute position measurement. The CAM312 chip combines data from both fine and coarse coils to deliver a full absolute position output.

4.1 Circuit Schematics, Type 6

The circuit for connecting a Type 6 sensor to the CAM312 chip is shown in Figure 9Figure 8 below. Component values are listed in Table 11.

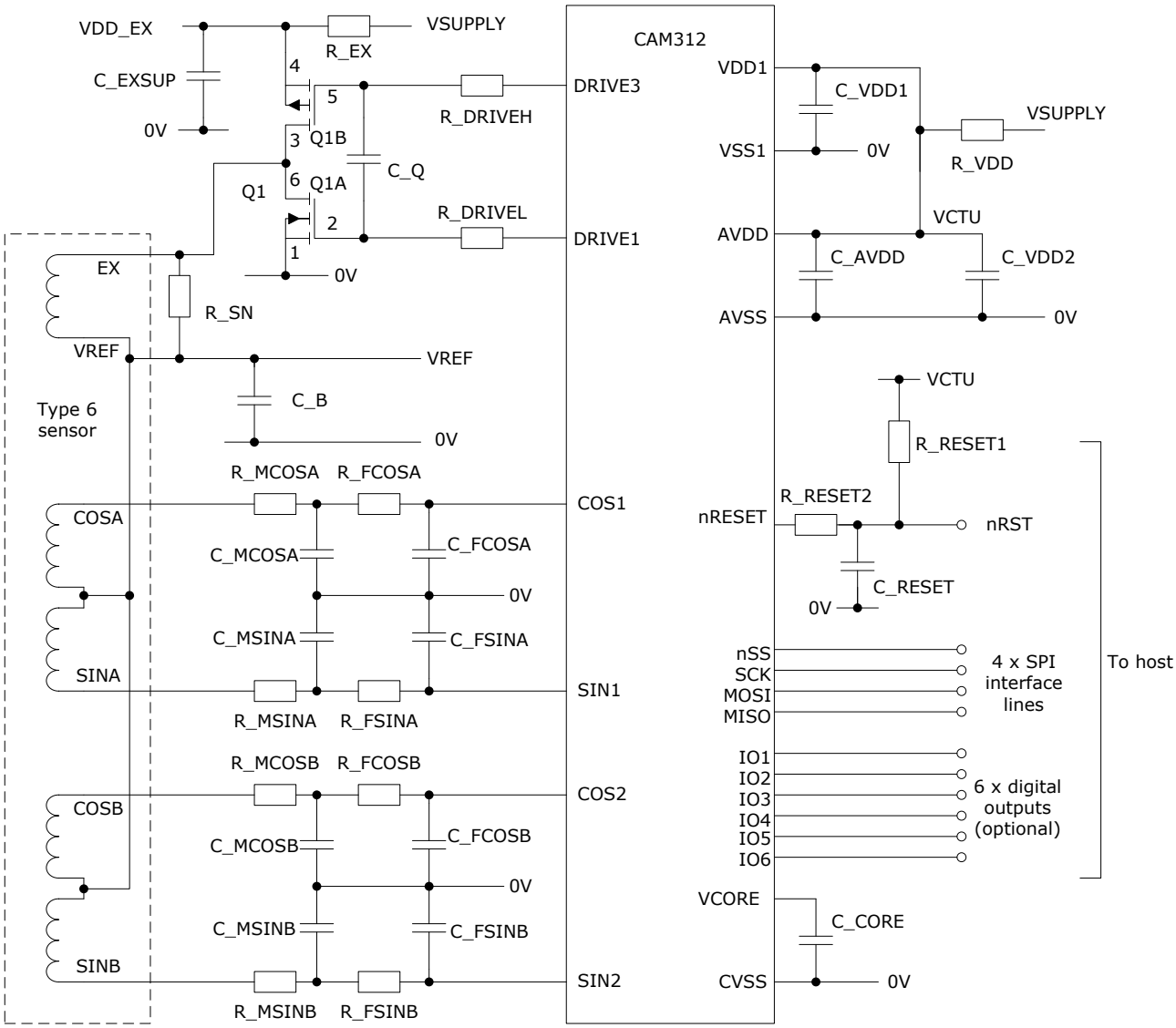


Figure 9 CAM312 connections, Type 6

The circuit design is very similar to that for Type 4 sensors, except for the additional filter components for processing signals from the sensor's coarse sensor coils COSB and SINB. Please refer to section 3.1 for a description of the circuitry.

Note that MISO is an open drain output when nSS is high, and requires a pull-up resistor of 4.7kΩ if this is not provided by the host.

4.2 Components Required, Type 6

Table 11 lists the component values and numbers required for the schematic of Figure 9.

Table 11 components required for Type 6 sensor connection

Circuit Ref	Value	Tolerance		Number required
		Grade A	Grade B	
R_RESET1	10kΩ	±5%		1
R_RESET2	470 Ω	±5%		1
R_VDD	1Ω	±5%		1
R_EX	4.7Ω	±5%		1
R_DRIVEL	1kΩ	±5%		1
R_DRIVEH	330Ω	±5%		1
R_SN	1kΩ	±5%		1
R_MCOS/SINA	150Ω	±0.1%	±1%	2
R_FCOS/SINA	150Ω	±0.1%	±1%	2
R_MCOS/SINB	150Ω	±5%		2
R_FCOS/SINB	150Ω	±5%		2
C_VDD1, C_AVDD	100nF	±10%		2
C_CORE, C_EXSUP, C_VDD2	10µF, ESR < 1 Ω	±20%		3
C_RESET	1nF	±10%		1
C_Q	1nF	±10%		1
C_MCOS/SINA	1nF	±1%	±5%	2
C_FCOS/SINA	1nF	±1%	±5%	2
C_MCOS/SINB	1nF	±10%		2
C_FCOS/SINB	1nF	±10%		2
C_B	4.7µF	±10%		1
Q	Si1016CX			1

The value of C_EXSUP may be increased in order to reduce peak supply current, see section 12.2.

The filter components connected to the CAM312 chip's sensor inputs are important for reproducibility. The table above includes two grades for them: A and B. Grade A yields the least system to system reproducibility error due to component differences. Grade B components are slightly more cost effective. Grade A and B reproducibility error is compared in Table 12.

Table 12 reproducibility error due to filter components, Type 6

Grade	Reproducibility due to filter components (% of SinLengthA)	Reproducibility examples	
		Rotary Type 6.5, SinLengthA=72°	Linear Type 6.12, SinLengthA = 31.6mm
A	±0.04%	±0.03°	±0.013mm
B	±0.2%	±0.14°	±0.063mm

Note that only the filter components connecting the fine sensor coils require tighter tolerance components (e.g. R_MCOS/SINA). Filter components connecting the coarse sensor coils do not require tight tolerance because coarse coil errors do not contribute to measurement results.

5 Type 2 Sensor Application

Type 2 sensors are for the precise measurement of linear position. Like Type 6 sensors, their additional precision comes from fine ("A") sensor coils whose signals repeat multiple times along the measuring path. This multiplies resolution and accuracy. Coarse ("B") sensor coils provide full absolute position measurement. The CAM312 chip combines data from both fine and coarse coils to deliver a full absolute position output.

5.1 Circuit Schematics, Type 2

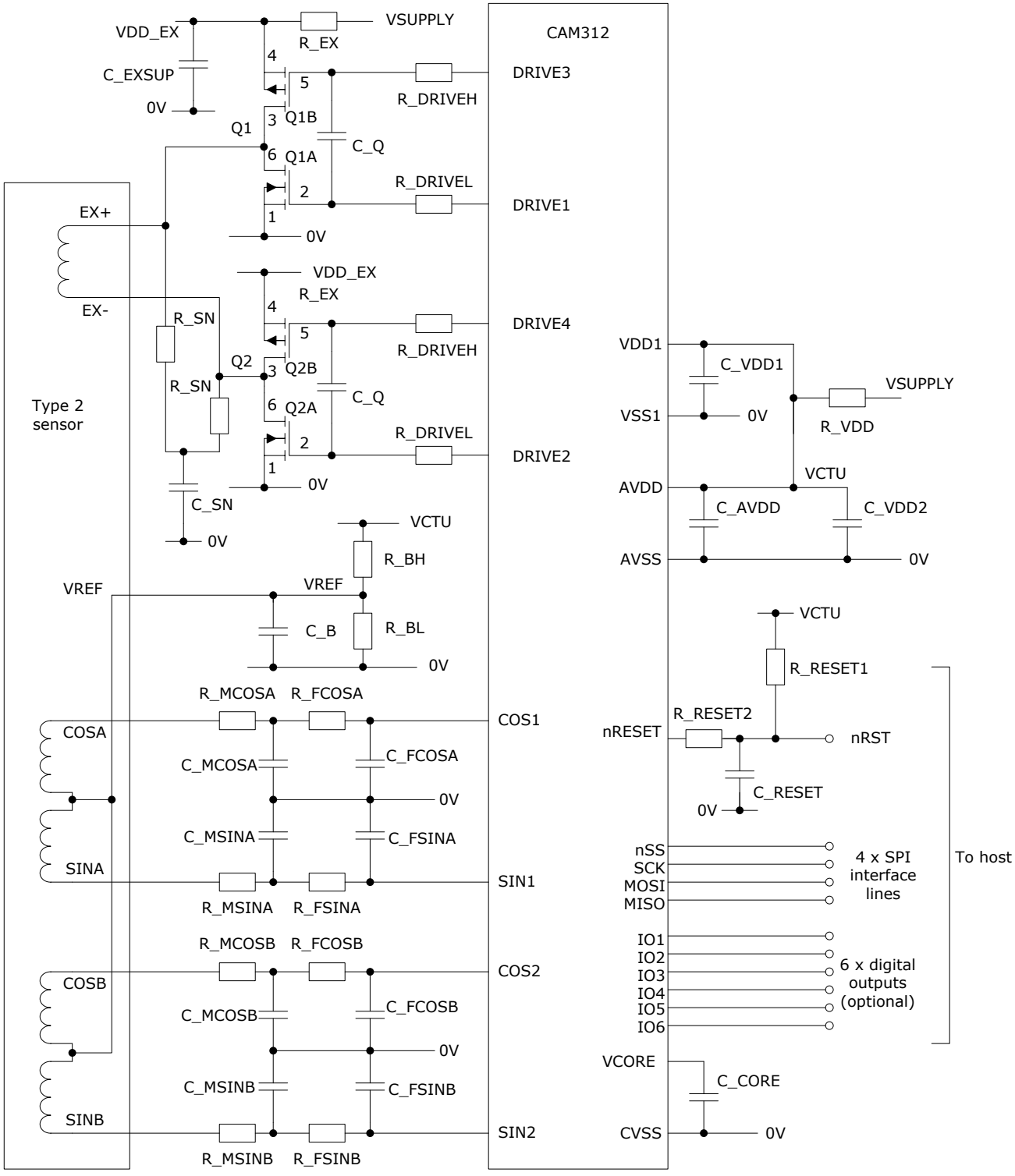


Figure 10 CAM312 connections, Type 2

The circuit for connecting a Type 2 sensor to the CAM312 chip is shown in Figure 10 above. Component values are listed in Table 13.

The difference between Type 2 and Type 6 sensor circuitry is that Type 2 uses a full bridge drive circuit for the excitation coil. This delivers a greater voltage across the excitation coil, allowing it to apply greater power which is generally required for Type 2 sensors due to their larger size.

The full bridge drive circuit comprises MOSFET pairs Q1 and Q2, and their associated gate drive components. The two resistors R_SN and capacitor C_SN absorb the energy in the excitation coil on the H bridge's transition from low to high impedance.

Resistors R_BH and R_BL provide a DC bias voltage VREF for the sensor coil inputs to the CAM312 chip. These bias resistors are not present in Type 4 and Type 6 circuitry. They are required for the Type 2 circuitry because the excitation process does not provide a suitable bias voltage, and the excitation coil does not share decoupling capacitor C_B.

If repeated measurements in low power mode is required (section 11.7) then the bias arrangement shown in Figure 10 is NOT suitable. Its recovery from power down is too slow. Please contact CambridgeIC for an alternative.

In other respects the circuit design is very similar to that for Type 4 and Type 6 sensors. Please refer to section 3.1 for a description of the circuitry.

Note that MISO is an open drain output when nSS is high, and requires a pull-up resistor of 4.7kΩ if this is not provided by the host.

5.2 Components Required, Type 2

Table 13 lists the component values and numbers required for the schematic of Figure 10.

Table 13 components required for Type 2 sensor connection

Circuit Ref	Value	Tolerance		Number required
		Grade A	Grade B	
R_RESET1	10kΩ	±5%		1
R_RESET2	470 Ω	±5%		1
R_VDD	1Ω	±5%		1
R_EX	1Ω	±5%		1
R_DRIVEL	1kΩ	±5%		2
R_DRIVEH	330Ω	±5%		2
R_SN	100Ω	±5%		2
R_MCOS/SINA	150Ω	±0.1%	±1%	2
R_FCOS/SINA	150Ω	±0.1%	±1%	2
R_MCOS/SINB	150Ω	±5%		2
R_FCOS/SINB	150Ω	±5%		2
R_BH, R_BL	10kΩ	±5%		2
C_VDD1, C_AVDD	100nF	±10%		2
C_CORE, C_EXSUP, C_VDD2	10µF, ESR < 1 Ω	±20%		3
C_RESET	1nF	±10%		1
C_Q	1nF	±10%		2
C_MCOS/SINA	1nF	±1%	±5%	2
C_FCOS/SINA	1nF	±1%	±5%	2
C_MCOS/SINB	1nF	±10%		2
C_FCOS/SINB	1nF	±10%		2
C_B	4.7µF	±10%		1
C_SN	10nF	±10%		1
Q1, Q2	Si1016CX			2

The value of C_EXSUP may be increased in order to reduce peak supply current, see section 12.2.

The filter components connected to the CAM312 chip's sensor inputs are important for reproducibility. The table above includes two grades for them: A and B. Grade A yields the least system to system reproducibility error due to component differences. Grade B is slightly more cost effective. Grade A and B reproducibility error is compared in Table 14.

Table 14 reproducibility error due to filter components, Type 2

Grade	Reproducibility due to filter components (% of SinLengthA)	Reproducibility examples, Linear Type 2.12	
		SinLengthA=31.2mm	SinLengthA=50.0mm
A	±0.04%	±0.13mm	±0.02mm
B	±0.2%	±0.063mm	±0.1mm

Note that only the filter components connecting the fine sensor coils require tighter tolerance components (e.g. R_MCOS/SINA). Filter components connecting the coarse sensor coils do not require tight tolerance because coarse coil errors do not contribute to measurement results.

6 Type 3 Sensor Application

Type 3 sensors are for the precise measurement of position. Like Type 2 and 6 sensors described above, Type 3 sensors include both fine ("A") and coarse ("B") sensor coils, to deliver precision without loss of absolute position. The resonator inside the target is energised with current applied to each coarse coil in turn. This is for sensor designs where a single excitation coil is not possible. Examples include linear sensors where the magnetic axis of the target is along the measuring direction ("Axial"). Axial sensors allow the target to rotate about this magnetic axis, and are therefore suited to level sensing applications and piston style linear sensors where the target must be allowed to rotate freely.

6.1 Circuit Schematics, Type 3

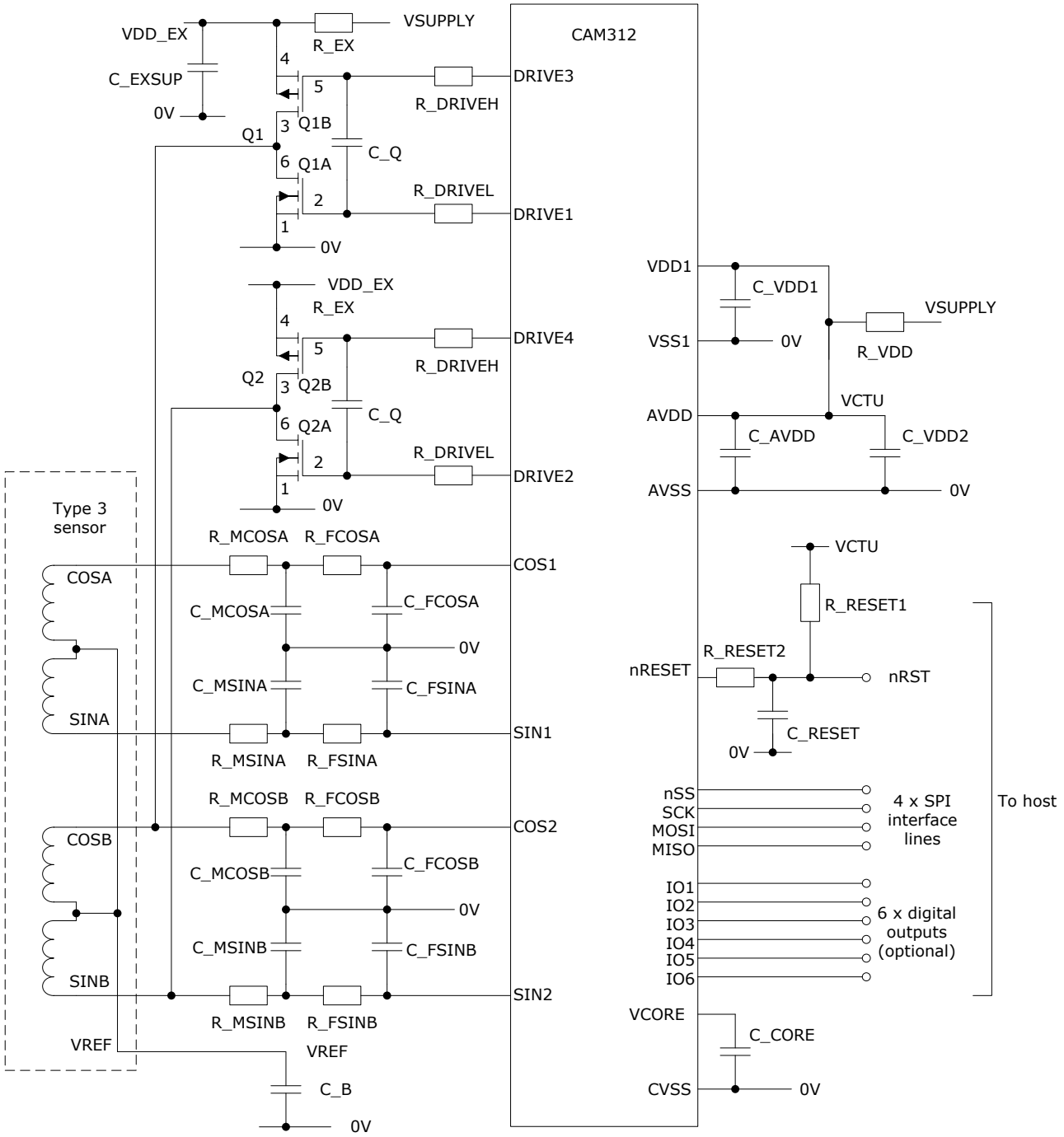


Figure 11 CAM312 connections, Type 3

The circuit design for Type 3 sensor connection is shown in Figure 11. Component values are listed in Table 15.

The basic circuitry for reset, excitation and inputs is similar to for a Type 4 sensor. Please refer to section 3.1 for more on the design and its operation.

There are two pairs of input stages, one for fine "A" COS and SIN coil inputs, and one for coarse "B" COS and SIN inputs. The coarse coils each have their own excitation half bridge which allows the CAM312 to power each coarse coil in turn, and then receive and detect resonator signal from the same coils.

6.2 Components Required, Type 3

Table 15 lists the component values and numbers required for the schematic of Figure 11.

Table 15 components required for Type 3 sensor connection

Circuit Ref	Value	Tolerance		Number required
		Grade A	Grade B	
R_RESET1	10kΩ	±5%		1
R_RESET2	470 Ω	±5%		1
R_VDD	1Ω	±5%		1
R_EX	4.7Ω	±5%		1
R_DRIVEL	1kΩ	±5%		2
R_DRIVEH	330Ω	±5%		2
R_MCOS/SINA	150Ω	±0.1%	±1%	2
R_FCOS/SINA	150Ω	±0.1%	±1%	2
R_MCOS/SINB	150Ω	±0.1%	±1%	2
R_FCOS/SINB	150Ω	±0.1%	±1%	2
C_VDD1, C_AVDD	100nF	±10%		2
C_CORE, C_EXSUP, C_VDD2	10µF, ESR < 1 Ω	±20%		3
C_RESET	1nF	±10%		1
C_Q	1nF	±10%		2
C_MCOS/SINA	1nF	±1%	±5%	2
C_FCOS/SINA	1nF	±1%	±5%	2
C_MCOS/SINB	1nF	±1%	±5%	2
C_FCOS/SINB	1nF	±1%	±5%	2
C_B	4.7µF	±10%		1
Q	Si1016CX			2

The value of C_EXSUP may be increased in order to reduce peak supply current, see section 12.2.

The filter components connected to the CAM312 chip's sensor inputs are important for reproducibility. The table above includes two grades for them: A and B. Grade A yields the least system to system reproducibility error due to component differences. Grade B components are slightly more cost effective. Grade A and B reproducibility error is compared in Table 16.

Table 16 reproducibility error due to filter components, Type 3

Grade	Reproducibility due to filter components (% of sensor's SinLengthA)	
	Type 3.2	**Type 3.9**
A	±0.06%	±0.04%
B	±0.3%	±0.2%

The SinLengthA value depends on the sensor, and is quoted in its datasheet. For a linear Type 3 sensor with Subtype 2, SinLengthA is usually slightly more than half of the measuring range. With Subtype 3, it is slightly more than a third of the measuring range, and so on.

7 Circuit Layout

This section contains recommendations for the PCB layout of the CAM312 circuitry, and applies to Type 4 sensor circuitry (section 3), Type 6 (section 4) and Type 2 (section 5).

7.1 Ground Plane and Capacitor Connections

It is recommended to use a PCB with at least 4 layers, one of which should be a solid ground plane below the CAM312. Conductors carrying signals not associated with the CAM312 chip should not be placed between the CAM312 chip and ground plane.

Connections from the CAM312 chip's VSS, AVSS and CVSS pins to the ground plane must be made with vias placed close to the chip. They should ideally also be connected together to the centre pad. Capacitors C_VDD1, C_AVDD and C_CORE must be kept close to the chip with short distances to ground connections. The same goes for C_B, C_FCOS1 and C_FSIN1, and C_FCOS2 and C_FSIN2 if used. Figure 12 illustrates recommended locations for these components, and their connections. Connections to the CAM312 chip's other pins are less critical.

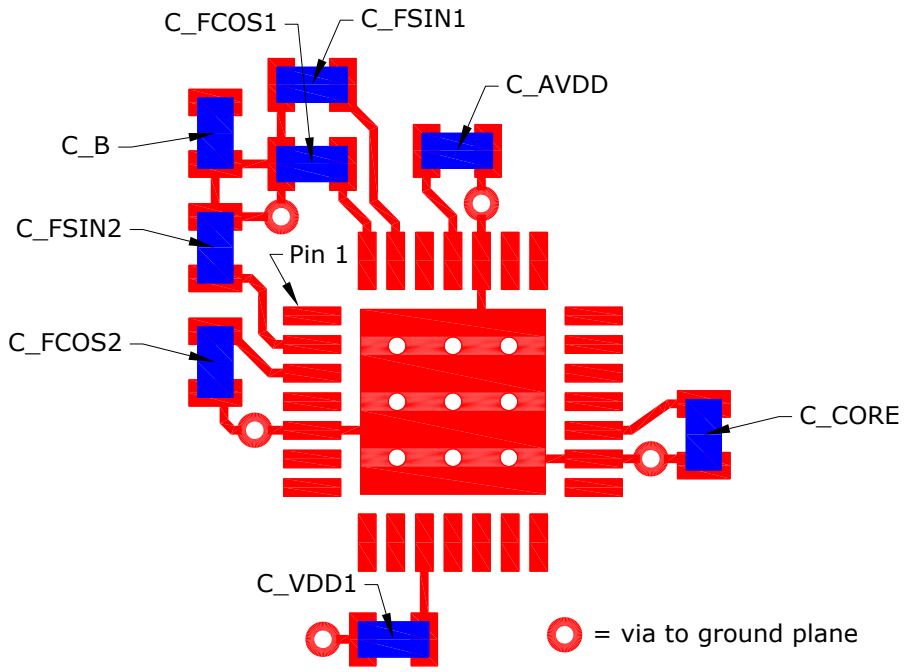


Figure 12 Recommended capacitor locations and wiring

7.2 Sensor Coil and Filter Network Connections

Figure 13 illustrates connections between sensor coils and the CAM312 circuit, including its filter components. When a sensor coil is connected to the CAM312 circuit, the traces and/or wires forming the connection make a loop. The loop formed by the COSA coil (or a Type 4 sensor 1 COS coil) connections is shaded as an example.

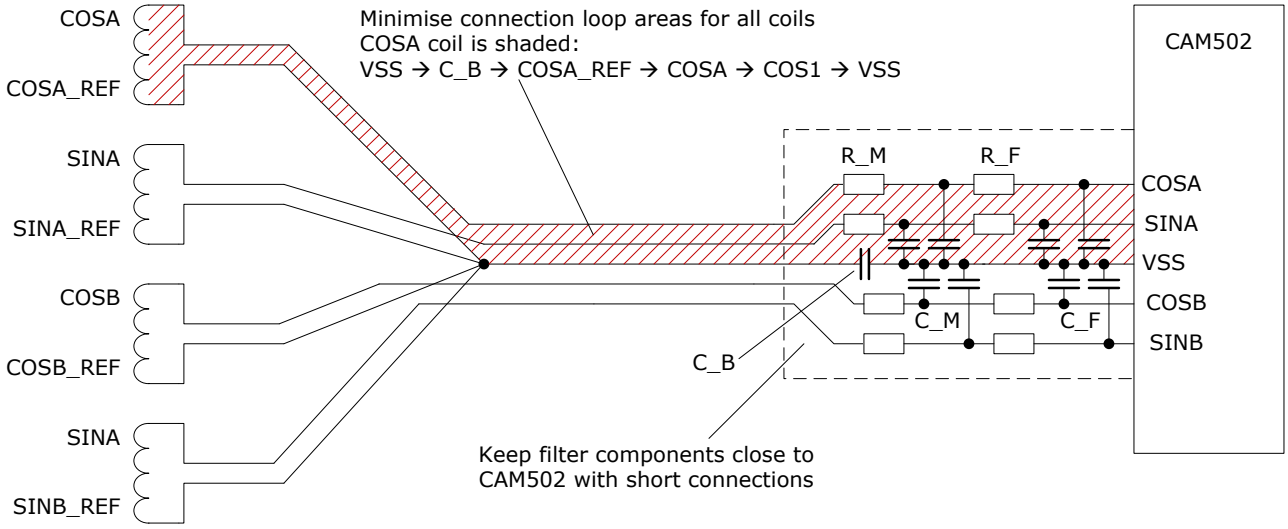


Figure 13 Sensor coil and filter connections to CAM312

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. Wires used for connection must be run in a tight bundle, on adjacent conductors in a ribbon cable or twisted.

Conductors which run for more than 100mm on a PCB should be arranged in coil pairs as in Figure 14 (a) or (b), or adjacent to a common VREF conductor as in Figure 14 (c) or (d).

Shorter conductors may be arranged above or below a ground plane, as in Figure 14 (e). In this case the conductors must avoid the outer 4mm of the ground plane, where the ground plane is less effective at shielding loops against signals from a nearby target and/or AC magnetic interference.

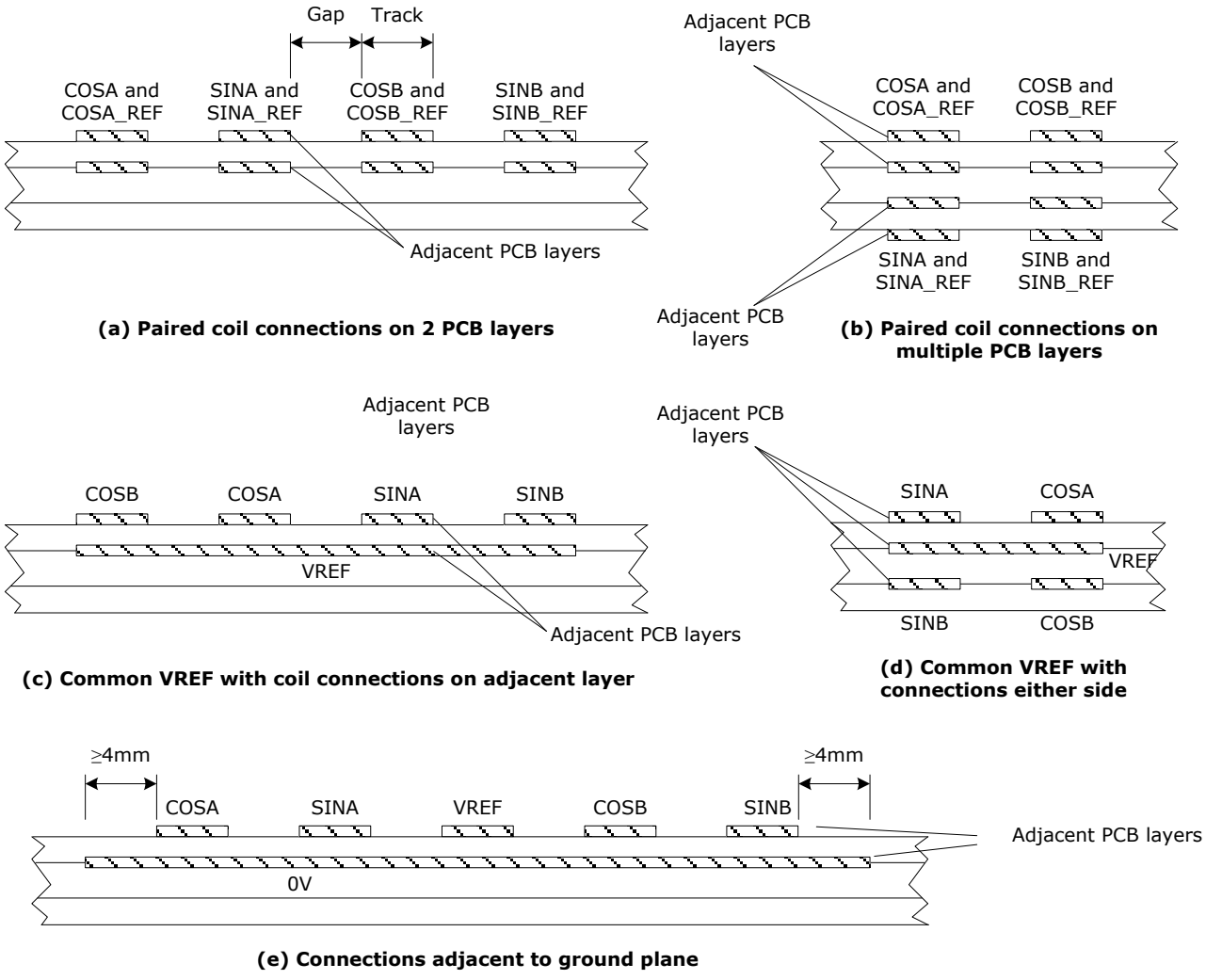


Figure 14 PCB conductor arrangements for sensor coils

Track and Gap figures should be minimised, providing connection resistance does not fall below 5Ω, or below 1Ω for a VREF connection that is common to two or more coils.

7.3 Excitation Circuit and Coil Connections

The CAM312 chip drives the gates of external MOSFET pair(s), which in turn drive current into the excitation coil to energise the resonator inside the target. Type 2 circuitry uses a circuit employing two MOSFET pairs configured as an H bridge. Type 4 circuitry (Figure 7) and Type 6 circuitry (Figure 9) use a single MOSFET pair per sensor configured as a half bridge. Type 3 circuitry uses two MOSFET pairs per sensor, each configured as a half bridge.

To keep the circuit efficient and to minimise emissions, the excitation circuit's decoupling capacitor C_EXSUP must be kept close to the MOSFET pair(s), and should be connected to them with fat traces and the minimum of trace lengths.

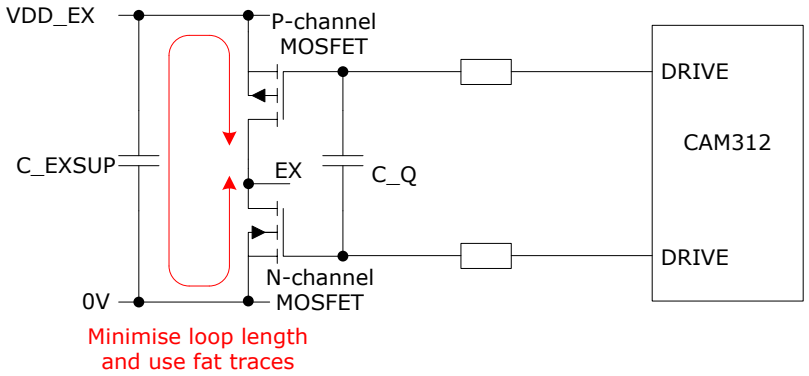


Figure 15 Excitation circuit layout

Each capacitor C_Q should be located near the MOSFET it is connected to.

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

The excitation coil connecting track widths should preferably be wide to minimise resistance. A loop resistance of less than 0.5Ω is preferable. PCBs with 1oz copper (34µm thick) have a surface resistivity of 0.5mΩ/square, so connections can be up to 1000 squares long. Table 17 shows how the minimum width of each trace in the pair of traces connecting the excitation coil varies with length.

Sensor and excitation traces may be run adjacent to one another. Type 4 and Type 6 sensors share a common VREF for both sensor and excitation coil, and the EX connection may be added to one side of the parallel connections shown in Figure 14 (c) or (d). It is more important to minimise sensor coil connection area than excitation, in any case that there is a conflict between the two.

Table 17 Minimum conductor widths on 1oz (34µm) PCB for ≤ 0.5Ω loop resistance

Length of excitation conductor pair	Minimum conductor width
100mm	0.2mm
200mm	0.4mm
400mm	0.8mm
800mm	1.6mm

8 Resonator Detection

8.1 Background

The CAM312 chip may be connected to different sensor types as described in the preceding sections. In each case the sensor couples with a moving target to measure its position. The coupling is inductive (AC magnetic fields). The target includes an electrical resonator comprising an inductor ("L") and capacitor ("C"). The system is illustrated in Figure 16.

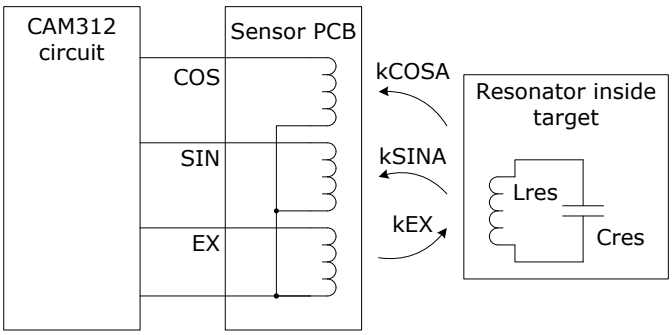


Figure 16 CAM312 connected to sensor coupled with resonator in target

A resonator can be characterized with two parameters: its resonant frequency "Fres" and quality factor "Q". The resonant frequency is the frequency of the resonance illustrated in Figure 5. The Q is a measure of how fast the resonance decays. Resonant inductive sensors generally return higher signal levels and work better with higher Q values, corresponding to slower decay.

The CAM312 chip energises the excitation coil to power the resonator at its resonant frequency, and then listens to the resulting fields in the COS and SIN senor coils. This process is described in section 1.1. The CAM312 chip is designed to lock onto and track the resonant frequency of inductively coupled resonators inside targets. The CAM312 chip measures the position of each sensor's target in turn. Each time, it also estimates its resonant frequency, providing there is sufficient signal amplitude. If the resonant frequency changes, the CAM312 chip changes its excitation frequency to match.

Following power on reset, or if the previous measurement was invalid, the CAM312 chip searches its tuning range for the target. Each measurement the CAM312 tries a different frequency until it achieves lock. This means it can take a few measurements before the CAM312 reports VALID for any sensor, as specified in Table 18.

Sensors and their targets are described in separate datasheets. Some targets include a prepackaged resonator, for example CambridgeIC's Standard Target part 013-1005. Other sensors work with targets that customers can build themselves, from inductors and capacitors. In this case the resonant frequency is given by Equation 1

Equation 1 Resonant Frequency

$$Fres = \frac{1}{2\pi\sqrt{Lres \times Cres}}$$

The inductor value Lres may be the value of a single inductor, or it may comprise two or more inductors connected together as in a "Type 6 Through Hole Rotary Sensor".

If the target and resonator are well away from metal and magnetic material then they are said to be in "free space". If not, the inductor value Lres may be modified by the presence of the material, and the resulting value of Fres will be similarly modified according to Equation 1. When designing targets and their resonators for use with the CAM312 chip, it is always the actual resonant frequency "seen" by the CAM312 chip that matters, taking into account the effect of metal and any magnetic material nearby.

The CAM312 chip is designed to work with resonators across a wide frequency range. This means that the effect of nearby metal and magnetic materials is easy to budget for, and that capacitor and inductor tolerances can easily be accommodated. When considering the effects of inductor and capacitor tolerances, the differential of Equation 1 is useful…

Equation 2 Change in resonant frequency due to inductor and capacitor changes

$$\frac{\Delta Fres}{Fres} = -0.5 \times \left(\frac{\Delta Lres}{Lres} + \frac{\Delta Cres}{Cres}\right)$$

For example if Lres has a tolerance of ±3% and Cres a tolerance of ±5%, the resulting tolerance of Fres is ±4%.

For more on frequency tolerances and selection of appropriate resonant frequency and tolerances, please refer to the CambridgeIC white paper "Resonant Frequency Centering".

8.2 Resonator Detection Specifications

Table 18 lists parameters relevant to the CAM312 chip's detection of resonators, and their values.

Table 18 Resonator Detection Specifications

Parameter	Value	Comment
Nominal Resonator Centre Frequency	187.5kHz	For a reported relative frequency of 0Hz with a nominal CAM312 chip
Resonator tuning range Relative to Nominal Resonator Centre Frequency	±11%	Across Operating Temperature Range -40°C to +125°C
	±10%	Across Operating Temperature Range -40°C to +150°C
Minimum Resonator Q-factor	25	For reliable position and frequency detection
Recommended Minimum Resonator Q-factor	50	
Minimum Amplitude	500	For reliable position and frequency detection
	1000	Recommended minimum for design purposes
Maximum number of measurements on a sensor with in-range target before VALID	6	Following a reset or resonator out of range

9 SPI Hardware

9.1 Overview

This section describes how data is written to, and read back from, the CAM312 chip, over its SPI interface. For communication with a host system, the CAM312 chip is always an SPI slave. All communication is initiated by the host.

There are two types of SPI transaction: a Register Access SPI and Checksum Access SPI. A Register Access SPI accesses one or more of the CAM312 chip's internal registers. It may either read or read-write the register(s). A Checksum Access SPI provides a checksum of the previous Register Access SPI's data, to assist with SPI error detection.

9.2 Data Transfer Method

The CAM312 chip always operates as an SPI slave device. The host device starts a data transfer by driving nSS low. It sends data to the CAM312 with the MOSI line, and provides the CAM312 with a serial clock line SCK. The CAM312 detects each MOSI bit on the rising edge of SCK. The CAM312 chip sends data back to the host with the MISO line. Bits change state on the falling edge of SCK, and the host should detect the state of MISO on each rising edge. This is commonly referred to as SPI Mode 0. The beginning and end of an SPI transaction is illustrated in Figure 17.

All SPI transactions MUST be bounded by the Slave Select (nSS) line being driven low at their start and being driven high at their end. The SPI interface will not function if nSS is tied permanently low.

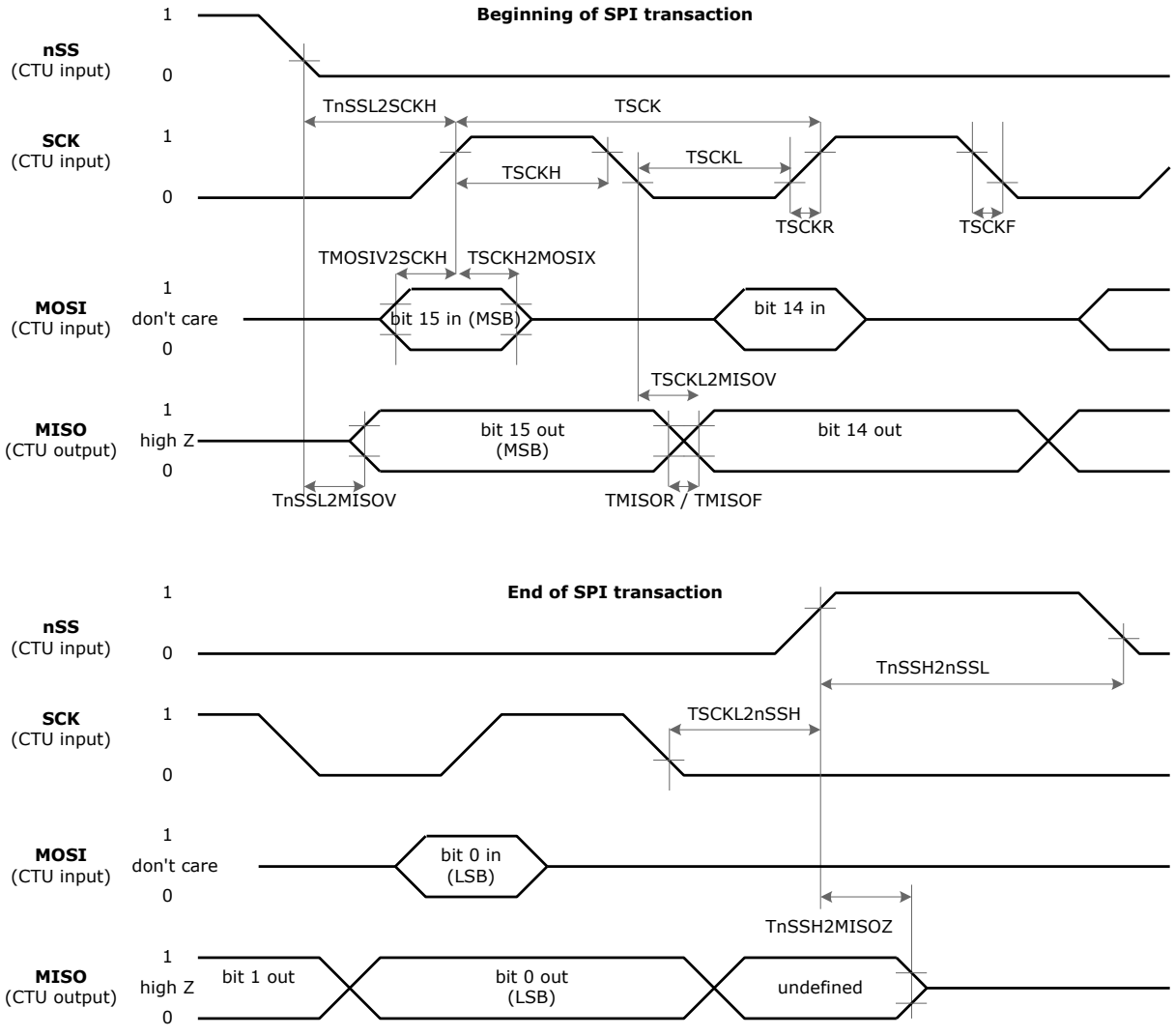


Figure 17 SPI Data Transfer

The timing parameters defined in this section apply to both Register Access SPI transactions and Checksum Access SPI transactions.

9.3 Register Access SPI

The host operates the CAM312 chip by writing to and reading from registers. There are a number of registers with different functions. Each register is 16 bits wide, and has its own 12 bit address.

The host must start each SPI transaction by clocking out 4 ACS bits field followed by 12 address bits on MOSI. The function of the ACS bits is defined in Table 14. The ACS bits define whether the current SPI transaction reads from registers (Read), or writes and reads data (Write Read).

Table 19 ACS bits definition

ACS[3:0]	Abbreviation	Access type
0x0	RS	Read
0xF	WRS	Write Read

When the host requests a Read, the next 16 bits clocked out of MISO after the address are the data contained in the register at the specified address. The CAM312 ignores the state of MOSI during data transfer. This transaction is illustrated in Figure 18.

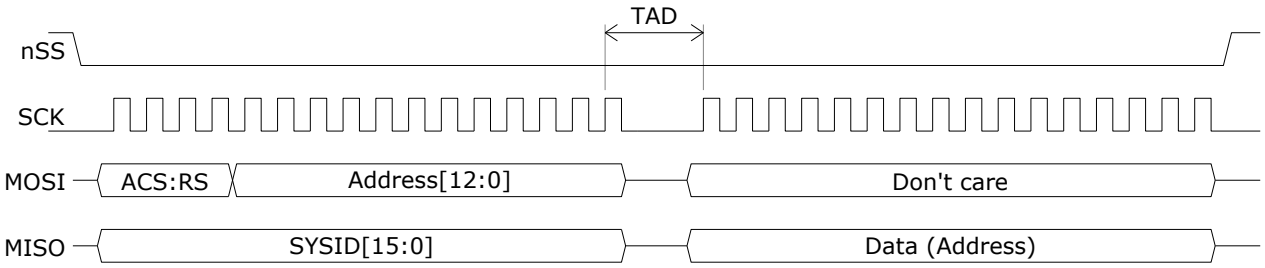


Figure 18 Read, single register

A Write Read operation is similar, except the host also updates the register contents with the new data that the host clocks out of the MOSI line after the address. The data returned on the MISO line is the register contents before the update. This transaction is illustrated in Figure 19.

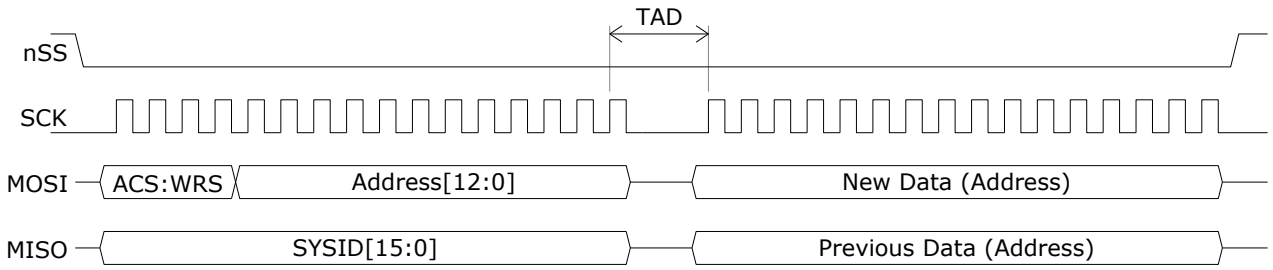


Figure 19 Write Read, single register

For both of these Register Access SP operations, a time TAD is required between the rising edge of SCK that clocks in the last address bit and the rising edge of SCK that clocks out the first data bit, illustrated in Figure 18 and Figure 19 above. TAD allows the CAM312 time to load its SPI transmit buffer with data from the correct address. This ensures that MISO data is available to the host before the next rising edge of SCK. Different minimum values apply depending on whether or not a measurement is in progress.

If a Register Access SPI transaction is being performed when a measurement is in progress, the required value of TAD is longer to allow the CAM312 to complete measurement tasks which are assigned a higher priority than handling SPI. A measurement is in progress if any sensor is set for continuous measurement (SCW:CONT=GO=1), or if a measurement has been started on any sensor (SCW:GO=1) and not yet finished (sample indicator still clear).

The CAM312 chip outputs the contents of its SYSID register as the first word of each Register Access SPI transaction. This defaults to 0xABCD, and may be changed by the host (section 10.9). Other values are possible when operating in

Bootloader mode, or as a result of errors (see section 11.12). It is recommended that the host checks the SYSID value read back against the expected value as a test for SPI communication integrity.

Instead of accessing a single register, the host may access a set of consecutive registers by extending the SPI transaction as illustrated in Figure 20. Data is transferred one register at a time, starting at the first specified address. Figure 20 illustrates a Multiple Write Read SPI transaction with the host providing ACS, address and new register data with the MOSI signal.

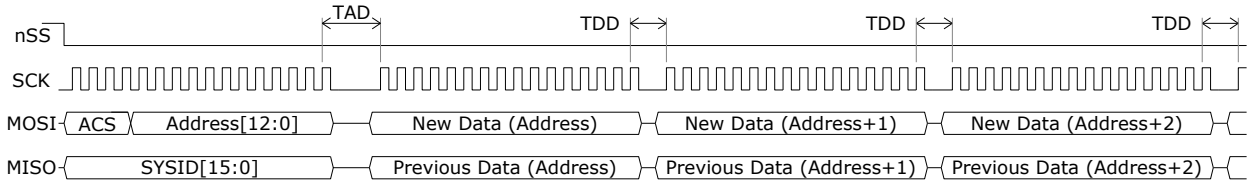


Figure 20 Multiple Register access

A Multiple Read SPI is similar. In this case, though, the host only provides the relevant ACS and address. Once the last address bit has been clocked into the CAM312, the state of MOSI does not matter for the remainder of the SPI transaction and is ignored by the CAM312.

Registers are arranged in blocks dedicated to system functions, and to each sensor connected, distinguished by the upper 4 bits of the address. Multiple register access must not span different blocks. It may only span either system registers or registers associated with a single sensor.

It is not essential to transfer complete 16 bit words into the CAM312. However it is recommended, since data from any incomplete word will be discarded.

Figure 20 defines SPI timing parameter TDD. TDD is measured between the last positive going edge of SCK of one data word to the first positive going edge of SCK of the next data word. The host must ensure that the TDD value it generates is greater than the minimum values specified in Table 20. Different minimum values apply depending on whether or not a measurement is in progress.

When the period of the host's SCK signal (TSCK) is shorter than TDD, pauses in the SCK signal are visible at data word boundaries, as illustrated in Figure 20. It is also possible for the SCK signal to run continuously without pauses, as illustrated in Figure 21. In this case the SCK clock period TSCK must exceed the specified value of TDD(min).

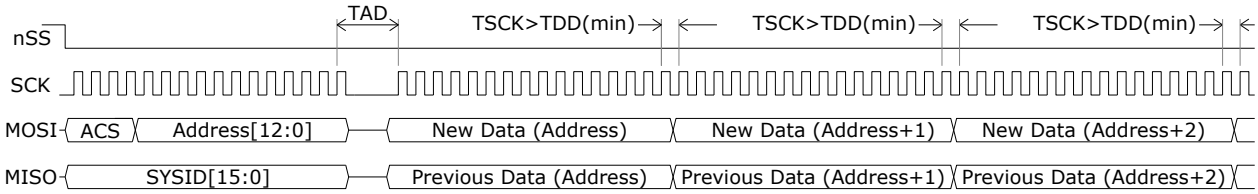


Figure 21 No pauses in SCK signal at data word boundaries

A minimum time is required between Register Access SPI transactions. This allows time for the CAM312 to perform any necessary reconfiguration before the next Register Access SPI occurs. This time is denoted TnSSHR2nSSLR and is illustrated in Figure 22.

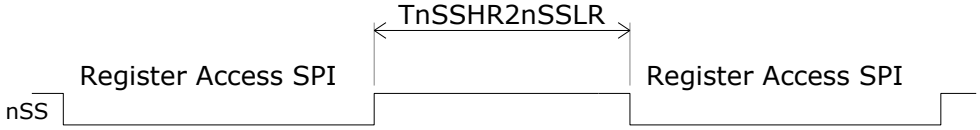


Figure 22 Time TnSSHR2nSSLR requirement

9.4 Timing of IO Changes Following a Register Access SPI

The host may clear a sensor's active Sample Indicator by writing 0 to the sensor's SIF bit in a write read transaction including the SCW register. The state of any User IO mapped to that Sample Indicator will change shortly after the end of the transaction, at a time TnSSH2IOclr defined in Figure 23. The maximum value of this timing parameter is specified in Table 20.

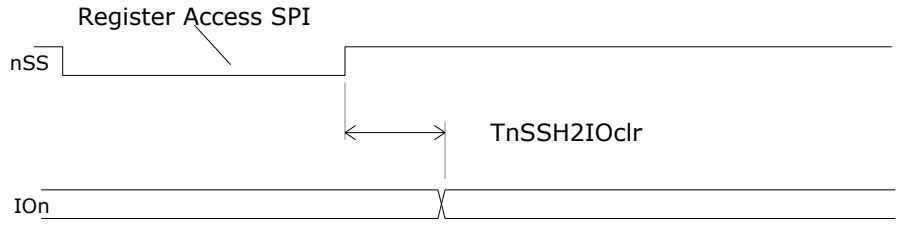


Figure 23 IO Change Following a Register Access SPI

Writes to the SYSIO or SIC registers may also change the state of one or more User IOs. If there is no measurement in progress, the maximum time to change to the new IO states is TnSSH2IOfn(max). This is measured from the end of the SPI transaction, like TnSSH2IOclr. If there is a measurement in progress the IO states may not change until after that measurement.

9.5 Checksum Access SPI

The CAM312 chip calculates a checksum of each Register Access SPI transaction's data. This may be read out immediately after each Register Access SPI transaction.

Reading the checksum is optional. If a checksum is not required, the host should ensure the SPI nSS line is held high for at least TnSSHR2nSSLR(min) after the SPI transaction. Please see Table 20 for the value of this timing parameter.

The CAM312 uses the timing between SPI transactions to distinguish between a Register Access SPI transaction and Checksum Access SPI transaction. If the nSS high time following a Register Access SPI transaction is between TnSSHR2nSSLC(min) and TnSSHR2nSSLC(max), then the then the following SPI is a Checksum Access SPI transaction. TnSSHR2nSSLC is illustrated in Figure 24 below. The minimum value depends on whether or not a measurement is in progress, defined in the same way as for TAD in section 9.3.

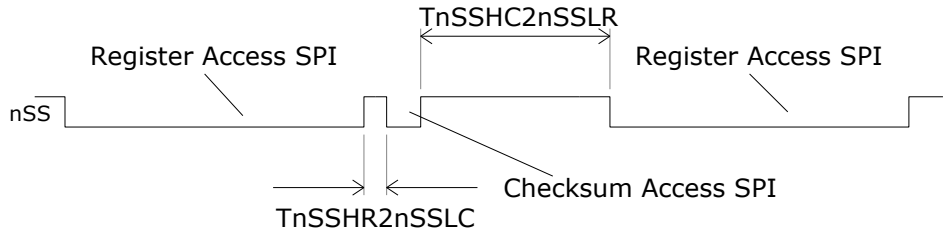


Figure 24 Time TnSSHR2nSSLC requirement

Note that TnSSHR2nSSLC(max) is less than TnSSHR2nSSLR(min), to help the CAM312 to make a clear distinction between Register Access SPI transactions and Checksum Access SPI transactions.

Once the host has read out all 16 bits of the checksum in a Checksum Access SPI transaction, it must take the nSS SPI signal high again. The nSS signal must remain high for at least a time TnSSHC2nSSLR(min) before going low again for the next Register Access SPI transaction. TnSSHC2nSSLR is defined in Figure 24 and specified in Table 20.

The checksum value is obtained by applying a CRC-16 algorithm to the MISO data output by the CAM312 in the preceding Register Access SPI transaction. That is, all complete data words which the CAM312 outputs on MISO, including the first word (SYSID) and the register contents which follow. When a checksum is required, the maximum length of the preceding Register Access SPI transaction is 16 words (SYSID plus 15 data words). The CRC algorithm is summarised in Equation 3, using standard shorthand notation.

Equation 3 CRC-16 Checksum

$$CRC\ Polynomial: x^{16} + x^{15} + x^2 + 1\ (0x8005)$$
$$CRC\ Initialisation: 0x0000$$

9.6 Interface Timing Specifications

Table 20 specifies values for the SPI timing parameters, including those defined in preceding subsections.

Table 20 Interface timing specifications

Parameter	Description		Min	Max	Units
TVALIDCHECK	Time for internal validity checks following reset		5	16	ms
TSCKL	SCK Input Low Time		30	-	ns
TSCKH	SCK Input High Time		30	-	ns
TSCK	SCK clock period		90	-	ns
TSCKR	SCK Input Rise Time		-	10	ns
TSCKF	SCK Input Fall Time		-	10	ns
TMISOR	MISO Rise Time (50pF load)		-	10	ns
TMISOF	MISO Fall Time (50pF load)		-	10	ns
TMOSIV2SCKH	MOSI Setup Time		30	-	ns
TSCKH2MOSIX	MOSI Hold Time		30	-	ns
TnSSL2MISOV	First MISO state valid after nSS low edge		-	80	ns
TSCKL2MISOV	MISO state valid after SCK low edge		-	20	ns
TnSSL2SCKH	nSS low to SCK edge		120	-	ns
TSCKL2nSSH	Last SCK edge to nSS high		90	-	ns
TnSSH2MISOZ	nSS high to MISO high Z		-	100	ns
TAD	Address end to first data	Measurement not in progress (1)	3	-	µs
		Measurement in progress (2)	8	-	µs
TDD	Delay between data words	Measurement not in progress (1)	0.24	-	µs
		Measurement in progress (2)	1.2	-	µs
TnSSHR2nSSLR	Time between end of one Register Access SPI and the beginning of the next		30	-	µs
TnSSHR2nSSLC	Window after SPI to read checksum	Measurement not in progress (1)	3	19	µs
		Measurement in progress (2)	8		µs
TnSSHC2nSSLR	Time between end of Checksum Access SPI and the next Register Access SPI		30	-	µs
TnSSH2IOclr	Time to clear sample indicator following an SPI that cleared it		-	25	µs
TnSSH2IOfn	Time to change IO function following SPI accessing SYSIO or SIC, measured from end of SPI or any measurement in progress		-	30	µs

Operation according to sections 11.3, 11.4 or 11.5 ensures there is no measurement in progress during the SPI transaction. This allows the host to operate with minimum SPI timings, marked (1) in Table 20. To use these timings, the host must not perform any Register Access SPI transactions after the one that sets any sensor's GO bit to 1 and that sensor's measurement completing, either indicated by the activation of that sensor's sample indicator or by waiting for the appropriate Minimum Measurement Time specified in Table 35.

The host must use longer SPI timings if the CAM312 may be performing a measurement at the same time as the SPI transaction, marked (2) in Table 20. These longer SPI times are essential in continuous mode (any sensor's SCW:CONT bit set to 1, as in section 0). The extra time allows the CAM312 chip to perform measurement functions and respond to SPI activity at the same time.

9.7 Total SPI Transaction Times

Table 21 is a guide to achievable SPI timings. Timing parameters are based on a host operating with fast SPI parameters including a bit rate of 10Mbit/s. The timing parameters selected are close to the minimum values of Table 20, but also allow for typical host tolerances and implementation.

Table 21 Register Access SPI transaction time examples

SPI timings	Registers Accessed	SPI transaction time
TSCK = 100ns (“10Mbit/s”) TnSSL2SCKH = 200ns TAD = 3µs (measurement not in progress) TDD = 300ns SCKL2nSSH = 200ns	SCW, PTEF, RESA, RESB, RESC (Type 4 results)	14µs
	SCW, PTEF, RESA, RESB, RESC, RESD, RESE, RESF (Type 2, 3 or 6 results)	20µs
	Any one register	7µs

10 Register Description

The host configures and controls the CAM312 chip by writing to its internal registers. It obtains status and measurement results by reading back from internal registers. This section describes the function of each of the CAM312 chip's registers.

Reading and writing to registers is done over the SPI interface, using the procedures described in section 9. Section 11 describes how to control the CAM312 chip to perform sensor position measurements.

The CAM312 chip's register map is arranged in distinct sections: a system control section and one section for each sensor. The arrangement is illustrated in Figure 25 and Figure 26.

The register map includes a number of registers and bit fields whose functions are not implemented in the CAM312, for compatibility with other CTU chips. For example the PTEF register controls position trigger behaviour in the CAM204 chip, but is not used in the CAM312.

The register map has been arranged to reduce SPI overheads once the whole system has been initialised following a reset. Once initialised, the host system will normally only need to access a small contiguous block of registers to control each sensor, which can be performed within a single SPI transaction.

	Address	Register
SENSOR 2	0x 2 1F	PTP8
	0x 2 1E	PTP7
	0x 2 1D	PTP6
	0x 2 1C	PTP5
	0x 2 1B	PTP4
	0x 2 1A	PTP3
	0x 2 19	PTP2
	0x 2 18	PTP1
	0x 2 17	PTAH
	0x 2 16	PTC87
	0x 2 15	PTC65
	0x 2 14	PTC43
	0x 2 13	PTC21
	0x 2 12	SIC
	0x 2 11	STYPE
	0x 2 10	DACCW
	0x 2 0F	DACLIMY
	0x 2 0E	DACLIMX
	0x 2 0D	DACPOSD
	0x 2 0C	DACPOSC
	0x 2 0B	DACPOSB
	0x 2 0A	DACPOSA
	0x 2 09	LEDCW
	0x 2 08	CONST
	0x 2 07	RESF
	0x 2 06	RESE
	0x 2 05	RESD
	0x 2 04	RESC
	0x 2 03	RESB
	0x 2 02	RESA
	0x 2 01	PTEF
	0x 2 00	SCW

	Address	Register	Description
SENSOR 1	0x 1 1F	PTP8	Pos Trig Position 8
	0x 1 1E	PTP7	Pos Trig Position 7
	0x 1 1D	PTP6	Pos Trig Position 6
	0x 1 1C	PTP5	Pos Trig Position 5
	0x 1 1B	PTP4	Pos Trig Position 4
	0x 1 1A	PTP3	Pos Trig Position 3
	0x 1 19	PTP2	Pos Trig Position 2
	0x 1 18	PTP1	Pos Trig Position 1
	0x 1 17	PTAH	PT Auto-Clear & Hysteresis
	0x 1 16	PTC87	Position Trigger 8&7 Control
	0x 1 15	PTC65	Position Trigger 6&5 Control
	0x 1 14	PTC43	Position Trigger 4&3 Control
	0x 1 13	PTC21	Position Trigger 2&1 Control
	0x 1 12	SIC	Sample Indicator Control
	0x 1 11	STYPE	Sensor Type
	0x 1 10	DACCW	DAC Control Word
	0x 1 0F	DACLIMY	DAC Limit Y
	0x 1 0E	DACLIMX	DAC Limit X
	0x 1 0D	DACPOSD	DAC Position D
	0x 1 0C	DACPOSC	DAC Position C
	0x 1 0B	DACPOSB	DAC Position B
	0x 1 0A	DACPOSA	DAC Position A
	0x 1 09	LEDCW	LED Control Word
	0x 1 08	CONST	Sensor Constants
	0x 1 07	RESF	Result Register F
	0x 1 06	RESE	Result Register E
	0x 1 05	RESD	Result Register D
	0x 1 04	RESC	Result Register C
	0x 1 03	RESB	Result Register B
	0x 1 02	RESA	Result Register A
	0x 1 01	PTEF	Pos Trig Enables/Flags
	0x 1 00	SCW	Sensor Control Word

System	0x 0 0F	SYSID	Device ID
	0x 0 0E	SYSVER	System Version Number
	0x 0 0D	BOOTVER	Bootloader Version Number
	0x 0 0C	SAVEKEY	Save Key
	0x 0 0B		System Identity
	0x 0 0A		(Reserved)
	0x 0 09		(Reserved)
	0x 0 08		(Reserved)
	0x 0 07		(Reserved)
	0x 0 06		(Reserved)
	0x 0 05		(Reserved)
	0x 0 04		(Reserved)
	0x 0 03	SYSDAC	System DAC Register
	0x 0 02	SYSIO	IO Pin Types
	0x 0 01	SYSI	Continuous Sample Interval
	0x 0 00	SYSCW	System Control Word

Figure 25 Register Map Overview

	Address	Description	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENSOR 1	0x 1 1F	Pos Trig Position 8	PTP8	PTP8 [15:0]															
	0x 1 1E	Pos Trig Position 7	PTP7	PTP7 [15:0]															
	0x 1 1D	Pos Trig Position 6	PTP6	PTP6 [15:0]															
	0x 1 1C	Pos Trig Position 5	PTP5	PTP5 [15:0]															
	0x 1 1B	Pos Trig Position 4	PTP4	PTP5 [15:0]															
	0x 1 1A	Pos Trig Position 3	PTP3	PTP3 [15:0]															
	0x 1 19	Pos Trig Position 2	PTP2	PTP2 [15:0]															
	0x 1 18	Pos Trig Position 1	PTP1	PTP1 [15:0]															
	0x 1 17	PT Auto-Clear & Hysteresis	PTAH	PTAUTOCLR	HYSTERESIS [14:0]														
	0x 1 16	Position Trigger 8&7 Control	PTC87	PT8ACT [1:0]		PT8DIR [1:0]		PT8MAP [3:0]				PT7ACT [1:0]		PT7DIR [1:0]		PT7MAP [3:0]			
	0x 1 15	Position Trigger 6&5 Control	PTC65	PT6ACT [1:0]		PT6DIR [1:0]		PT6MAP [3:0]				PT5ACT [1:0]		PT5DIR [1:0]		PT5MAP [3:0]			
	0x 1 14	Position Trigger 4&3 Control	PTC43	PT4ACT [1:0]		PT4DIR [1:0]		PT4MAP [3:0]				PT3ACT [1:0]		PT3DIR [1:0]		PT3MAP [3:0]			
	0x 1 13	Position Trigger 2&1 Control	PTC21	PT2ACT [1:0]		PT2DIR [1:0]		PT2MAP [3:0]				PT1ACT [1:0]		PT1DIR [1:0]		PT1MAP [3:0]			
	0x 1 12	Sample Indicator Control	SIC									SAUTOCLR [1:0]		SCTRL [1:0]		SMAP [3:0]			
	0x 1 11	Sensor Type	STYPE									SUBTYPE [3:0]				TYPE [3:0]			
	0x 1 10	DAC Control Word	DACCW	DACNV [15:8]								DACEN	-	-	DACADDR [2:0]			DACOP [1:0]	
	0x 1 0F	DAC Limit Y	DACLIMY	DACLIMY[15:0]															
	0x 1 0E	DAC Limit X	DACLIMX	DACLIMX[15:0]															
	0x 1 0D	DAC Position D	DACPOSD	DACPOSD[15:0]															
	0x 1 0C	DAC Position C	DACPOSC	DACPOSC[15:0]															
	0x 1 0B	DAC Position B	DACPOSB	DACPOSB[15:0]															
	0x 1 0A	DAC Position A	DACPOSA	DACPOSA[15:0]															
	0x 1 09	LED Control Word	LEDCW	LEDTHRESHOLD [11:0]												LEDMAP [3:0]			
	0x 1 08	Sensor Constants	CONST	-	-	-	-	-	-	-	-	FCSEL [3:0]				ABSSEL [3:0]			
	0x 1 07	Result Register F	RESF	RESF [15:0]															
	0x 1 06	Result Register E	RESE	RESE [15:0]															
	0x 1 05	Result Register D	RESD	RESD [15:0]															
	0x 1 04	Result Register C	RESC	RESC [15:0]															
	0x 1 03	Result Register B	RESB	RESB [15:0]															
	0x 1 02	Result Register A	RESA	RESA [15:0]															
	0x 1 01	Pos Trig Enables/Flags	PTEF	PT8E	PT7E	PT6E	PT5E	PT4E	PT3E	PT2E	PT1E	PT8F	PT7F	PT6F	PT5F	PT4F	PT3F	PT2F	PT1F
	0x 1 00	Sensor Control Word	SCW	SENSOR [3:0]				PIE	PIF	INCE	INCF	SIE	SIF	NEW	VALID		TRIG	CONT	GO
System	0x 0 0F	Device ID	SYSID	SYSID [15:0]															
	0x 0 0E	System Version Number	SYSVER	SYSVER [15:0]															
	0x 0 0D	Bootloader Version Number	BOOTVER	BLVER [15:0]															
	0x 0 0C	Save Key	SAVEKEY	SAVEKEY [15:0]															
	0x 0 0B	System Identity	CTUID	CTUID [15:0]															
	0x 0 0A	(Reserved)																	
	0x 0 09	(Reserved)																	
	0x 0 08	(Reserved)																	
	0x 0 07	(Reserved)																	
	0x 0 06	(Reserved)																	
	0x 0 05	(Reserved)																	
	0x 0 04	(Reserved)																	
	0x 0 03	System DAC Register	SYSDAC	DACCAL	-	-	-	-	-	CPOL	CPHA					DACFORMAT [2:0]			DACSON
	0x 0 02	IO Pin Types	SYSIO	-	-	-	-	INT6AH	INT6DO	INT5AH	INT5DO	INT4AH	INT4DO	INT3AH	INT3DO	INT2AH	INT2DO	INT1AH	INT1DO
	0x 0 01	Continuous Sample Interval	SYSI	SYSI [15:0]															
	0x 0 00	System Control Word	SYSCW	RESET	BOOTLOAD	SAVE	FACTORY												PWRDN

Figure 26 Register Map Detail

10.1 SYSCW: System Control Word

SYSCW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	RESET	BOOTLOAD	SAVE	FACTORY	-	-	-	-	-	-	-	-	-	-	-	PWRDN
Access	R/W	R/W	R/W-1	R/W-1	R	R	R	R	R	R	R	R	R	R	R	R/W

Factory default value = 0x0000

Writing a 1 to the RESET bit will result in a device reset at the end of the SPI transaction. Registers will return to their default values and measurements will be aborted.

Writing a 1 to the PWRDN bit puts the CAM312 chip into a low power state. See section 11.6 for how to use this feature, and relevant timings.

Writing a 1 to the BOOTLOAD bit resets the CAM312 chip at the end of the SPI transaction. When the CAM312 chip comes out of reset, it will remain in Bootloader Mode (section 14) until the next reset.

Writing a 1 to the SAVE bit, together with 0x0C1C to the SAVEKEY register, will result in the current register contents being made the Configurable Defaults (section 0). Writing a 1 to the FACTORY bit, together with 0x0C1C to the SAVEKEY register, will result in restoration of the factory default register values. After the SAVE or FACTORY operations the CAM312 chip will reset.

10.2 SYSI: System Interval Register

SYSI	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x001	SYSI [15:0]															
Access	R/W															

Factory default value = 0x000A

The CAM312 includes an option for the CAM312 to take measurements continuously, without the host having to initiate each one over SPI. The SYSI register controls the time between the start of each sensor measurement. The time is the value of SYSI multiplied by 100µs, and is approximate.

If Sensor 1 and Sensor 2 are both configured for continuous operation, each measurement on sensor 1 is immediately followed by a measurement on sensor 2. SYSI controls the time between the start of each measurement on sensor 1.

Table 22 Example SYSI values

SYSI	Time between start of measurements	Comments
0	Each measurement immediately follows the previous one	For continuous measurement at fastest rate
5	500µs	Smallest practical value for regularly timed measurements on a single Type 4 sensor, or a Type 6 sensor with INCE=1
7	700µs	Smallest practical value for regularly timed measurements on a Type 6 sensor with INCE=0
10	1000µs	Smallest practical value for regularly timed measurements on two Type 4 sensors
23	2300µs	Smallest practical value for regularly timed measurements on a Type 3 sensor
100	10ms	100 per second
200	200ms	Maximum tested interval

10.3 SYSIO: System IO Configuration

SYSIO	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x002	-	-	-	-	INT 6AH	INT 6DO	INT 5AH	INT 5DO	INT 4AH	INT 4DO	INT 3AH	INT 3DO	INT 2AH	INT 2DO	INT 1AH	INT 1DO
Access	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Factory default value = 0x0000

The SYSIO register controls each IO pin's behaviour. Each pin may be configured for active low or active high with the coding of Table 30. Independently, each pin may be configured for open drain or digital with the coding of Table 31.

Table 23

INTnAH	Function
0	Active Low
1	Active High

Table 24

INTnDO	Function
0	Open Drain
1	Digital Output

10.4 SYSDAC: System DAC Register

SYSDAC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x003	DACCAL	-	-	-	-	-	CPOL	CPHA	NVMODE [1:0]		-	-	DACFORMAT [2:0]			DACSON
Access	R/W	R	R	R	R	R	R/W	R/W	R/W		R	R	R/W			R/W

Factory default value = 0x0000

The SYSDAC register controls how the CTU interfaces to an external Digital to Analog Converter (DAC), if present. This function is detailed in section 13. The host should set the DACSON bit if an external DAC is connected (Table 25). DACFORMAT[2:0] must be set to match the DAC's data format (Table 40). NVMODE[1:0] defines how the DAC output(s) behave when its matching sensor's last measurement was not valid (Table 43). CPOL and CPHA must be set to match the DAC SPI interface's mode (Table 26). DACCAL allows the host to set the DAC data directly for calibration purposes, if set (Table 27).

Table 25

DACSON	Function
0	No DAC output(s)
1	Control external DAC output(s)

Table 26

CPOL	CPHA	DAC SPI Mode
0	0	0
0	1	1
1	0	2
1	1	3

Table 27

DACCAL	Function
0	Normal DAC operation
1	DAC output forced, for calibration

10.5 CTUID: CTU Identity

CTUID	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00B	CTUID[15:0]															
Access	R															

Default value = 0x030C

This CTU Identity register may be used to read the identity of the connected CTU chip: 0x030C corresponds to the CAM312.

10.6 SAVEKEY: Save Key

SAVEKEY	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00C	SAVEKEY[15:0]															
Access	R/W															

Default value = 0x0001

This system register is used to prevent accidental changes being made to register defaults. For the SAVE and FACTORY operations to take effect, its value should be set to 0x0C1C immediately beforehand.

10.7 BOOTVER: Bootloader Version Number

BOOTVER	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00D	Bootloader Version Number [15:0]															
Access	R															

Factory default value = 0x0202

The BOOTVER register contains the fixed revision number for the CAM312 chip's Bootloader Code, and is read-only.

10.8 SYSVER: System Version Number

SYSVER	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00E	System Version Number [15:0]															
Access	R															

Factory default value = 0x0104 ("V1.4")

The SYSVER register contains the version number of the CAM312 chip's Application Code, and is read-only. It is updated when new code is successfully updated using the bootloader (section 14).

10.9 SYSID: System Device ID

SYSID	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00F	SYSID[15:0]															
Access	R/W															

Factory default value = 0xABCD

The System Device ID is a host-writeable word, and appears as the first word of every SPI transaction (see section 9.3). It is recommended that the host verifies this value against the expected one, to assist detection of communication errors.

The SYSID may be used to identify different SPI slave devices connected to the same host. Any value may be chosen; it is of no significance to the CAM312 chip. However it is strongly recommended to avoid 0x0000, 0xFFFF and any of the possible Status Codes (section 11.12), since these may also result from communication errors or bootloader operation.

The Device ID may be used to detect or verify that a CAM312 chip reset has occurred. The host should write a new SYSID value, and any subsequent reset will cause SYSID to return to its default value.

10.10 SCW: Sensor Control Word

SCW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn00	SENSOR[3:0]				PIE	PIF	INCE	INCF	SIE	SIF	NEW	VALID	-	-	CONT	GO
Access	R				R	R	R/W	R	R/W	R/W0	R/W0	R	R	R	R/W	R/W

Factory default value = 0xn000

The registers described above relate to system functions. The Sensor Control Word, and the other registers described below, relate to sensors connected to the CAM312 chip. More than one sensor may be connected, depending on sensor type, and these registers are distinguished by bits 8 to 11 of the address. Two Type 4 sensors may be connected, and the addresses of their SCW registers are 0x100 and 0x200.

The host can start a measurement on a sensor by setting its GO bit to 1. If the host also sets the CONT bit to 1 the CAM312 chip will measure that sensor continuously. In this case, the sample interval between measurements active sensors is configured with SYSI, see section 10.2. If a sensor's CONT bit is set to 0, setting its GO bit to 1 will result in a single measurement. The CAM312 chip will clear GO to 0 upon completion.

The VALID bit indicates that the last measurement result for the sensor was valid. The CAM312 sets the VALID flag when it determines the channel's target to be in range. The meaning of in range depends on the sensor and target, but typically means that it is physically aligned within specification and its resonant frequency is within specified limits.

The NEW flag indicates that new measurement data is available for the sensor. The CAM312 chip sets NEW to 1 when a measurement is completed. It is recommended that the host clears NEW back to 0 when reading result registers, using a multiple register write/read SPI transaction spanning SCW and the relevant results registers. That way, the host can unambiguously verify that the results it has collected from the CAM312 chip are new.

The SIF bit is the sample indicator flag, and SIE is the sample indicator enable bit. Sample indicators can be configured to activate an IO when a new, or a new valid, sensor measurement is available. When sample indicators are suitably configured (see section 10.18), the SIE bit controls whether sample indicators are enabled. Once activated, the host may clear a sample interrupt by writing a 0 to SIF (see also section 11.4 concerning timing).

INCE and INCF are for sensor Types 2 and 6, which support Incremental Mode. The INCE bit controls whether Incremental Mode is allowed. The INCF bit is a flag that indicates when an incremental measurement was actually performed. See section 11.9 for details.

SENSOR[3:0] equals the sensor number. The host may check that SENSOR equals the requested sensor number, to ensure that measurement results are associated with the expected sensor.

PIE, PIF

10.11 PTEF

This register is not implemented n the CAM312.

10.12 RESA…RESF: Results Registers

RESA … RESF	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn02 … 0xn07	RESA [15:0] … RESF [15:0]															
Access	R															

Factory default values = 0x0000

A sensor's results registers contain the results of the last measurement to complete on that sensor. The CAM312 chip supports different sensor types. The interpretation of result registers depends on the sensor type as described in Table 28.

Table 28 Interpretation of measurement results

Register	Sensor Type	
	4	**2, 3 and 6**
RESA	CtuReportedPositionI16	CtuReportedPositionI32 high word
RESB	AmplitudeU16	CtuReportedPositionI32 low word
RESC	RelativeFrequencyI16	AmplitudeAU16
RESD	Not used	AmplitudeBU16
RESE	Not used	RelativeFrequencyI16
RESF	Not used	BAPositionMismatchI16

CtuReportedPosition may be converted into measurement units with Equation 4 below. Sin Length is a characteristic of the sensor, and is quoted in sensor datasheets. Type 2 and 6 sensors have fine and coarse pitches. Sin Length should be set to Fine Pitch for use in Equation 4.

Equation 4 Interpreting position result

$$ReportedPosition = \frac{CtuReportedPosition}{65536} \times SinLength$$

Reported Position is a signed quantity which is nominally 0 when the Target Origin is aligned with the Sensor Origin. This is usually at the centre point of the sensor's travel.

CtuReportedPosition will be forced to 0 following an invalid measurement when the target is out of range. The host can use the VALID bit of the SCW register to distinguish this condition from zero Reported Position.

Amplitude is a measure of inductive signal strength and influences system performance, particularly resolution. Type 2 and 6 sensors have two Amplitude results, one based on readings from fine sensor coils (AmplitudeA) and one for coarse coils (AmplitudeB). The coarse value is usually significantly less than the fine. References to Amplitude in a sensor's datasheet are to Amplitude A unless otherwise noted.

Relative Frequency is the CAM312 chip's approximate measurement of the frequency difference in Hz between the target and the CAM312 chip's Centre Frequency, whose nominal value is specified in Table 18.

BA Position Mismatch is specific to Type 2 and 6 sensors. It is scaled in the same way as Reported Position using Equation 4. When results are from Incremental Mode operation (INCF=1), BA Position Mismatch is the difference between the current and previous reported position. In absolute mode (INCF=0), BA Position Mismatch is the difference between position readings from the fine and coarse sensor tracks. In each of these cases BA Position Mismatch is an indication of system health. Small values are healthy, while values close to SinLength/2 indicate potential for error in position measurement caused by skipping a fine period (section 11.9).

10.13 LEDCW: LED Control Word

LEDCW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn09	LEDTHRESHOLD[11:0]												LEDMAP[3:0]			
Access	R/W												R/W			

Factory default value = 0x0000

User IOs can be configured to activate an LED when the result of a sensor's last measurement was VALID. Each sensor's LEDMAP[3:0] controls which user IO is activated by that sensor. The mapping is as in Table 37. The IO will toggle at approximately 2Hz for Amplitudes less than LEDTHRESHOLD, and it will activate continuously for Amplitudes greater or equal to LEDTHRESHOLD. Please refer to section 13.11 for details on using this feature.

10.14 DACPOSA…D: DAC Transfer Function Position Controls

RESA … RESF	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn0A … 0xn0D	DACPOSA[15:0] … DACPOSD[15:0]															
Access	R/W															

Factory default values = 0x0000

DACPOSA, DACPOSB, DACPOSC and DACPOSD are used when an external DAC is connected. They help define the relationship between measured sensor position and the DAC output mapped to the sensor (the DAC Transfer Function). They are in signed, 16-bit position units. DACPOSA and DACPOSB define the beginning and end of the up slope. DACPOSC and DACPOSD define the beginning and end of the down slope. Please see section 13.7 for more details.

10.15 DACLIMX and DACLIMY: DAC Transfer Function Level Controls

DACLIMX,Y	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn0E, 0xn0F	DACLIMX[15:0] … DACLIMY[15:0]															
Access	R/W															

Factory default values = 0x0000

DACLIMX and DACLIMY are used when an external DAC is connected. They help define the relationship between measured sensor position and the DAC output mapped to the sensor (the DAC Transfer Function). They are in unsigned, 16-bit DAC Code units. DACLIMX defines the lower DAC value, and DACLIMY the upper DAC value. Please see section 13.7 for more details.

10.16 DACCW: DAC Control Word

DACCW	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn10	DACNV[15:8]								DACEN	-	-	DACADDR[2:0]			DACOP[1:0]	
Access	R/W								R/W	R	R	R/W			R/W	

Factory default value = 0xn000

DACCW, the DAC Control Word, is used when an external DAC is connected. When enabled (a sensor's DACEN=1) the CAM312 CTU chip sends data to an external DAC each time that sensor's measurement completes.

Where the CTU is connected to a DAC with multiple outputs, a sensor's DACADDR[2:0] bits control which DAC output the sensor controls. DACOP[1:0] contains DAC output control bits. Their function is DAC-specific, and usually includes a power-down mode. Please see section 13.3 for details.

DACNV[15:8] helps control the DAC output when the sensor's last measurement was not valid, and is in DAC Code units. The exact behaviour depends on NVMODE[1:0]. Please see section 13.8 for details.

10.17 STYPE: Sensor Type Register

STYPE	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn11	-	-	-	-	-	-	-	-	SUBTYPE[3:0]				TYPE[3:0]			
Access	R	R	R	R	R	R	R	R	R/W (all sensors)				R/W (sensor1 only)			

Factory default value = 0x0004

The CAM312 chip is designed to operate with different types of sensor, see Table 1. To operate different sensor Types, the host must write the appropriate number to sensor 1's TYPE field before any position measurements. Other sensors' TYPE fields are read only. They will read either 0 if measurement on this sensor is not allowed or sensor 1's TYPE if measurement is allowed.

Some sensors have Subtypes. For example "Type 6.5" is Type 6, Subtype 5. For each sensor used, the SUBTYPE bits should be set to the sensor's Subtype.

Table 29 lists example sensors of sensor configurations, and the settings required for each STYPE register.

Table 29 STYPE settings required for example configurations

Connected sensors	Value to write to STYPE sensor 1
One Type 4 sensor connected to sensor 1 (COS1, SIN1)	0x0004
One Type 4 sensor connected to sensor 2 (COS2, SIN2)	0x0004
Two Type 4 sensors	0x0004
Type 6.5 sensor (Type 6, Subtype 5)	0x0056
Type 2.12 sensor (Type 2, Subtype 12)	0x00C2
Type 3.2 sensor (Type 3, Subtype 2)	0x0023

10.18 SIC: Sample Indicator Control Register

SIC	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn12	-	-	-	-	-	-	-	-	SAUTOCLR[1:0]		SCTRL[1:0]		SMAP[3:0]			
Access	R	R	R	R	R	R	R	R	R/W		R/W		R/W			

Factory default value = 0x0000

Sample indicators signal whether the CAM312 has new, or new valid, measurement results available for a sensor. The function is configured with SCTRL[1:0], see Table 30.

Table 30 SCTRL function

SCTRL[1:0]	Function
0	No sample indicators
1	Set on any new position
2	Set if VALID flag set
3	Reserved

Sample indicators can be configured to control the CAM312 chip's digital IOs. SMAP[3:0] configures which of the available IOs is used for the sensor's sample indicator. See Table 31 for how this bit field is encoded.

Table 31 SMAP function

SMAP[3:0]	IO pin mapping
0	Not mapped to an IO
1	IO1
2	IO2
3	IO3
4	IO4
5	IO5
6	IO6
7-15	reserved

Sample indicators can be reset by the host over the SPI interface by writing a 0 to the sample indicator flag SIF, see section 10.10. Alternatively, they may be "auto-cleared". SAUTOCLR[1:0] controls sample indicator auto-clear behaviour. Table 32 defines how this bit field is encoded.

Table 32 SAUTOCLR function

SAUTOCLR[1:0]	Auto clear behaviour
0	Autoclear OFF
1	Clear during measurement
2	Clear after measurement
3	Reserved

Sample indicators can be used to generate an IO whose state reflects whether the sensor's last measurement was VALID or not. Set SCTRL to 2 ("on valid") and SAUTOCLR to 2 ("after measurement"). The IO will only change state if there is a change in VALID.

Sample indicators can be used to create an active edge on an IO each time a measurement has completed…

- If each measurement is initiated by the host over SPI by setting GO=1, then the host can clear SIF during that same transaction. In this case the IO will clear after the SPI transaction (subject to timing parameter TnSSH2IOch of Table 20). In this case use SAUTOCLR=0 (factory default value).
- If two sensors are mapped to the same IO then configure each sensor to "clear during measurement" (SAUTOCLR=1).
- If sensors are sampling continuously (CONT=GO=1) then configure each such sensor to "clear during measurement" (SAUTOCLR=1). See section 0.

10.19 PTAH: Hysteresis

PTAH	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn17	-	HYSTERESIS[14:0]														
Access	R	R/W														

Factory default value = 0x0000

HYSTERESIS[14:0] sets the level of hysteresis applied to DAC output. Adding hysteresis can be helpful in some applications, because it can be used to stabilise the DAC output code in the presence of measurement noise. Its function is described in section 13.6. It is scaled in the same way as CtuReportedPositionI16.

11 System Operation

11.1 Reset, Validity Check and Reading Versions

The CAM312 CTU chip is internally reset following power on, when nRST is toggled (section 2.6) or when the host writes a 1 to the RESET bit in the SYSCW register (section 10.1).

Following a reset, the CAM312 performs internal validity checks, as illustrated in the flowchart of Figure 27.

Validity checks include a checksum of the CAM312 chip's FLASH memory contents, including Application Code. The time taken for these checks is TVALIDCHECK, as specified in Table 20. For normal operation, the host should allow at least TVALIDCHECK(MAX) between a reset and the first SPI communication with the CAM312, to avoid interrupting the validity checks.

The CAM312 has a set of 4 registers that contain version numbers, from CTUID (address 0x000B, see section 10.5) to SYSVER (0x000E, section 10.9). It is recommended that the host reads these 4 registers as the first SPI transaction after any reset. If validity checks passed successfully, the CAM312 will return the SYSID register contents (factory default 0xABCD) as the first word, followed by the version numbers.

If the CAM312 did not pass its validity checks it will return 0x10AD as the first word. This will usually be because Application Code is absent. Please refer to section 14 for details of loading code using the CAM312 chip's bootloader.

The CAM312 chip will also return 0x10AD as the first word if validity checks were interrupted by an SPI transaction before they completed.

Once the CAM312 chip has completed its validity checks and the host has optionally read out version numbers, the host can configure the CAM312 chip for taking measurements.

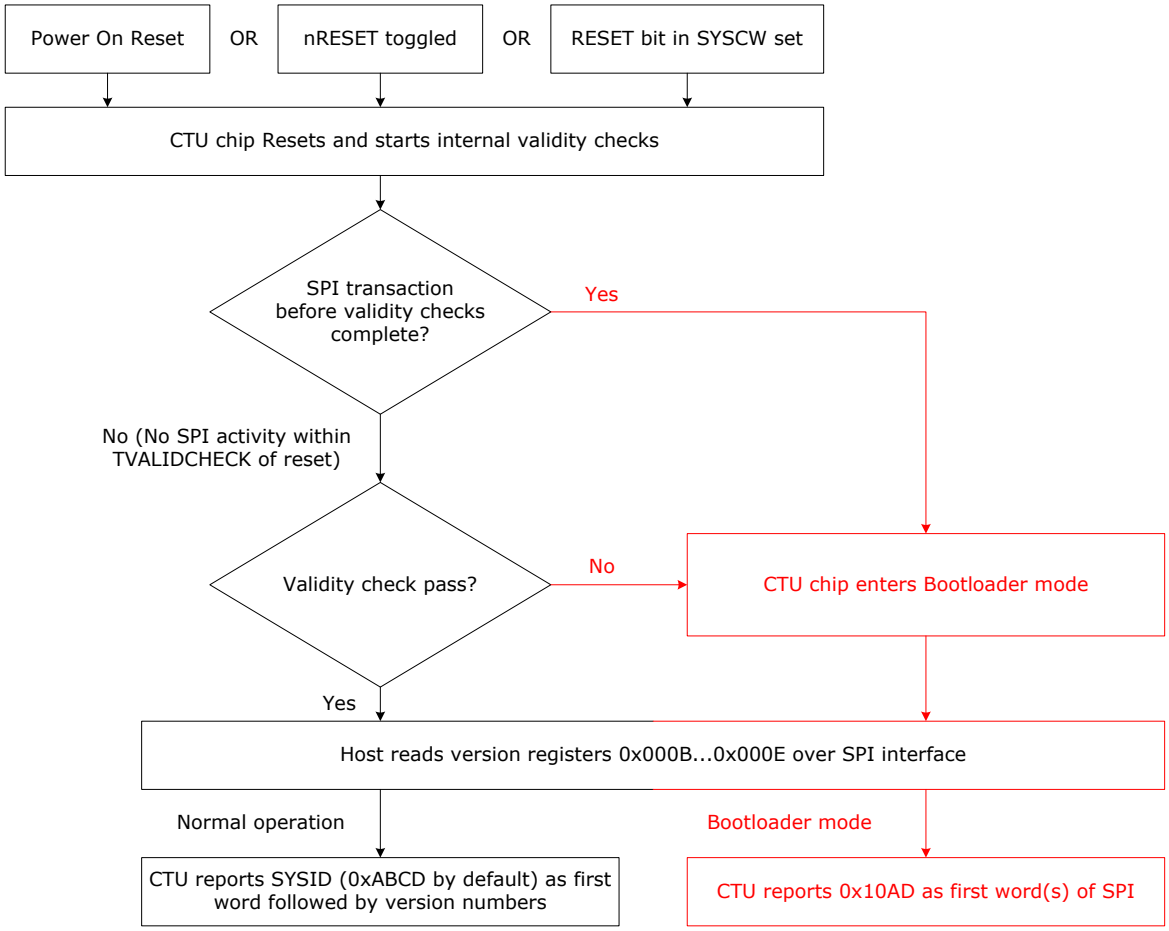


Figure 27 CAM312 reset and validity checking process

11.2 Configuration

Before taking measurements, the host must configure the CAM312. This is done by writing appropriate register contents to registers used to configure the required functions. Table 33 lists registers used for configuration, when they are needed, a summary of their function and where to find a detailed description.

Table 33 Registers to configure before measurement

Registers	When to use	Description	See...
SYSID	Optional	The host may change the SYSID value from its default value following a reset. This can help distinguish different CAM312 chips connected to the same SPI bus. It can also be used to detect an unexpected CAM312 reset.	Section 10.9
STYPE sensor 1	Essential	Tells CAM312 what type of sensor is connected to either or both sensor 1 and sensor 2	Section 10.17
SYSIO	If IOs used	Configures each IO for open drain or digital, and active high or active low	Section 10.3
SIC sensor 1	If sensor 1 sample indicator required	Tells CAM312 which IO to use for sensor 1 sample indicator, when to set and when to clear	Section 10.18
SIC sensor 2	If sensor 2 connected and sample indicator required	Tells CAM312 which IO to use for sensor 2 sample indicator, when to set and when to clear	
SYSI	Optional	Time between measurements operating in continuous mode (CONT=1)	Section 10.2

It is recommended to use write read single register SPI to access each register, as described in Figure 19. After each SPI transaction, the nSS SPI line must remain high for long enough for the CAM312 to complete each configuration step. The minimum high time is parameter TnSSHR2nSSLR(min) which is specified in Table 20.

Configuration registers may be accessed in any order.

SPI write read access returns the previous register settings, as they were before the SPI transaction. To check the new register settings are in place, the host may perform SPI read transactions.

11.3 Repeated Single Shot Measurement, One Sensor

In most applications the host needs regular measurement results from the CAM312. The simplest mode of operation is to repeat a single SPI transaction, which both starts a measurement (GO=1) and reads the results of the previous measurement. This process is illustrated in Figure 28.

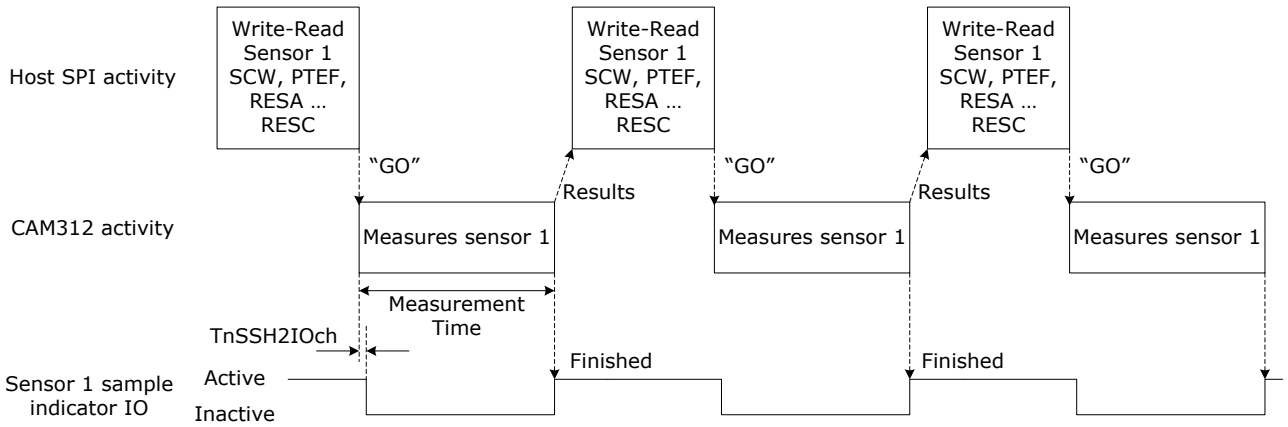


Figure 28 Repeated Single Shot Measurement, One Sensor

The host should use a multiple register access SPI transaction as illustrated in Figure 20. The first register to access is the sensor's SCW. If there is one Type 2 or 6 sensor connected, only sensor 1's SCW register may be accessed. If

there are two Type 4 sensors connected, the host may access either sensor 1 or sensor 2's SCW register, to perform measurements on sensor 1 or sensor 2. To alternate measurements between two sensors please refer to section 11.4 below.

The SCW register includes the GO bit which must be set to 1 each time so that the CAM312 performs another measurement. The CONT bit must be 0, so that CAM312 measurements are only initiated by the host and do not happen asynchronously. It is recommended to clear the NEW bit to 0, so that the host can identify whether the next measurement results are really new ones. The SIE bit must be set if the sensor's sample indicator is required. The SIF bit must be cleared to 0 so that the sample indicator returns to its inactive state following the SPI transaction ready for the next measurement. For Type 2 or Type 6 sensors, the INCE bit may be set to allow incremental mode , see section 11.9. The location of each bit within the SCW register is shown in section 10.10.

The number of words in the SPI transaction depends on how many results are needed. It is strongly recommended to read all results registers for the sensor Type in use (section 10.12). For example, if the sensor is Type 4, the SPI transaction should span SCW, PTEF, RESA, RESB and RESC. The PTEF and results registers are all read only, so it does not matter what data the host sends over the SPI MOSI line following the SCW register. 0x0000 is recommended for simplicity.

Once the SPI transaction has ended, the CAM312 starts a measurement, and clears any sample indicator IO mapped to the sensor. The time taken to clear the sample indicator is TnSSH2IOch, whose maximum value is specified in Table 20.

New measurement results are available for the host to read once the measurement has completed. The maximum time required for the measurement depends on sensor Type and operating mode. This Measurement Time is measured from the end of the SPI transaction setting GO to when measurements are available, as shown in Figure 28. Values are specified in Table 35. If used, a sample indicator IO will activate when new measurements are available. If not, the host should wait for at least the Maximum Measurement Time before performing another SPI transaction.

The host may then perform another SPI transaction to set GO again for the next measurement, and to read the results of the measurement just completed. These measurement results are found in the SCW register and in the results registers.

The SCW register includes the VALID bit, which indicates whether or not the measurement was VALID. It is important for the host to read the VALID bit, because results registers only contain valid data when VALID is set. It may take a number of measurements before the VALID bit is set, when the target is in range, see section 8.

The CAM312 chip sets the NEW bit each time a sensor's measurement completes. If the host reads NEW=0 then this indicates that there has been no new measurement since the previous SPI transaction, when NEW was cleared to 0. This could be caused by allowing insufficient time for the measurement to complete.

For sensor types which allow incremental mode, INCF indicates whether the measurement was actually an incremental one (1) or whether it was actually absolute (0), see section 11.9.

If valid, results registers may be interpreted according to section 10.12.

11.4 Repeated Single Shot Measurement, Two Sensors

The CAM312 chip can measure two Type 4 sensors. The host may take alternating measurements from two sensors with the process illustrated in Figure 29. Once a measurement has completed on sensor 1, the host may perform an SPI transaction reading sensor 2 results and setting sensor 2 to GO again. Once a measurement has completed on sensor 2, the host may perform an SPI transaction reading sensor 1 results and setting sensor 1 to GO again.

Please refer to section 11.3 for details of the SPI transactions required. The only difference between the two sensors is the address of the SCW register. Note that the CONT bit must still remain 0.

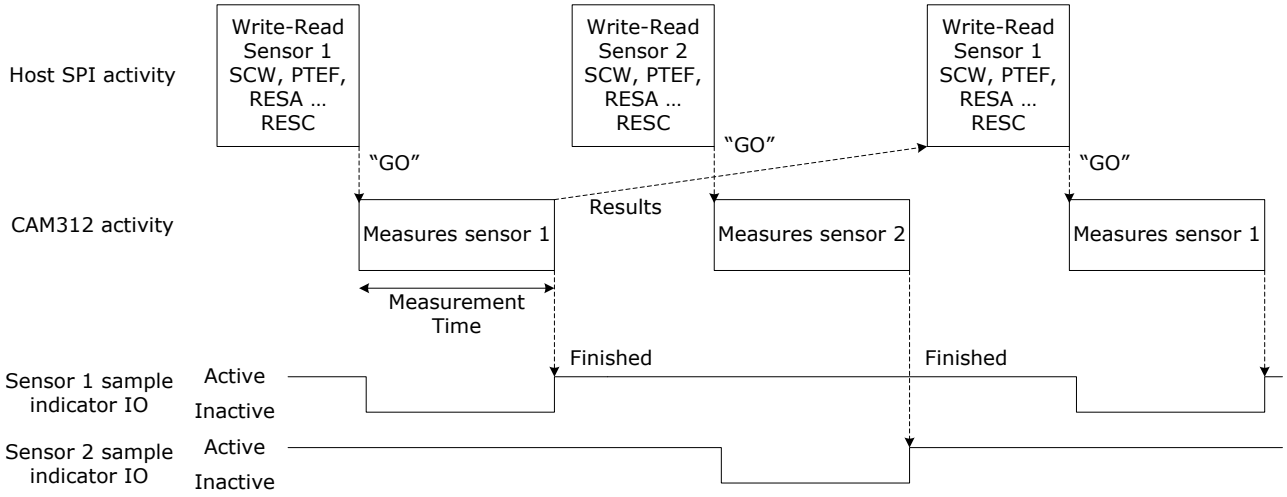


Figure 29 Repeated Single Shot Measurement, two sensors

Figure 29 illustrates two sample indicators, one for each sensor. The host can respond to the activation of sensor 1's sample indicator with a sensor 2 SPI, and the activation of sensor 2's sample indicator with a sensor 1 SPI.

Another option is to map both sensor sample indicators to the same IO, illustrated in Figure 30. Each time the IO activates the host can perform the next SPI transaction. In this case the SAUTOCLR bits of the SIC register must be set to "clear during measurement" as detailed in section 10.18, because clearing SIF only clears each sensor's sample indicator and not the other's. Note that the active to inactive IO edge no longer coincides with the end of each SPI transaction. Host devices will usually detect the inactive to active transition of the sample indicator by using that edge to activate a host device interrupt. If instead the host polls the state of the sample indicator, the host must first check the sample indicator has gone inactive following each SPI transaction, before detecting its transition to active.

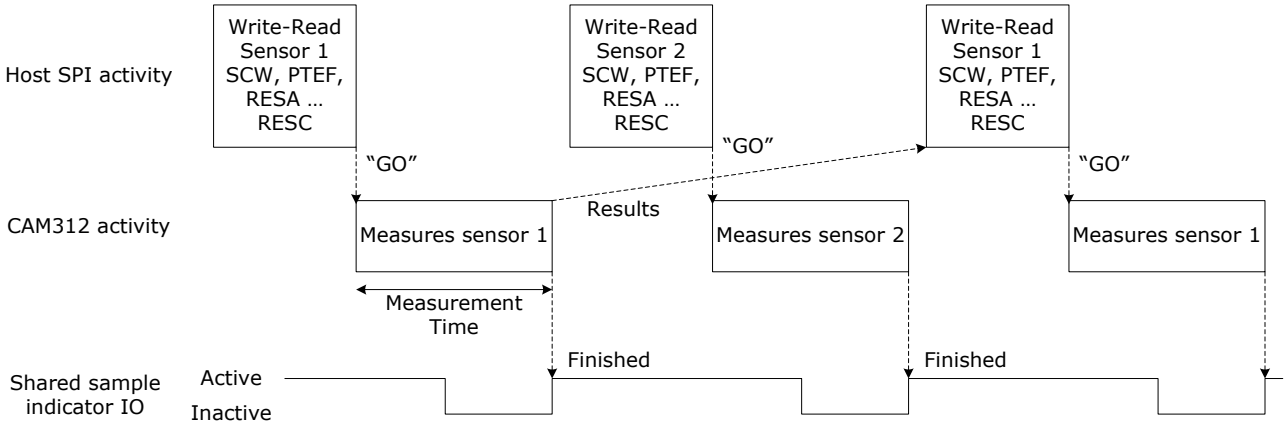


Figure 30 Repeated Single Shot Measurement, two sensors sharing a sample indictor IO

11.5 Isolated Single Shot Measurement

The measurement processes of sections 11.3 and 11.4 only require one SPI transaction per measurement. Each SPI transaction starts a measurement on a sensor, and reads back the measurement results from the preceding measurement on that sensor. If the host only requires occasional measurements, or measurements are taken at a low sample rate, there can be a long time interval between the CAM312 completing a measurement and the host reading the results.

This interval can be eliminated by performing two SPI transactions for each measurement: one to kick off the measurement and another to read results when they are ready. This process is illustrated in Figure 31.

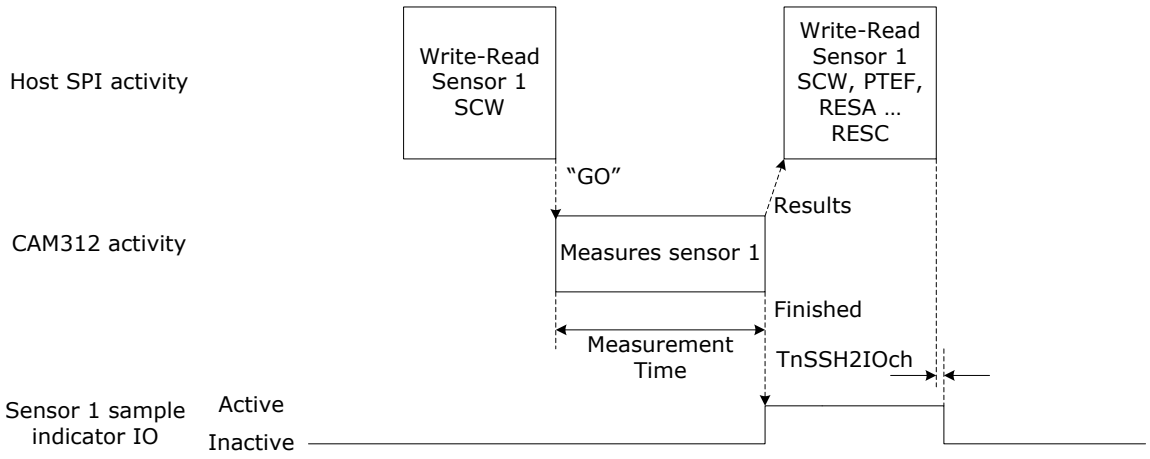


Figure 31 Isolated Single Shot Measurement

The SCW register contents used to initiate the measurement should be set up in the same way described in 11.3, including GO=1, CONT=0, NEW=0, SIE and INCE.

The SPI transaction used to read results should span SCW, PTEF and the results registers as before. It should be a write read transaction with the SCW register's GO=0, CONT=0 and NEW=0.

The host interprets results in the same way, including SCW register's VALID bit and the results registers themselves.

If two Type 4 sensors are connected, the process for an isolated single shot measurement can be applied to both sensors, one after the other.

11.6 Entering and Exiting Power Down Mode

This section explains how to use low power mode, so that the CAM312 draws a low supply current (section 12.1).

To put the CAM312 chip into low power mode, the host should write a "1" to the PWRDN bit in the SYSCW register over SPI, see section 10.1. Supply current will then reduce to the "PWRDN=1" levels given in Table 37 within TnSSHR2nSSLR(min) given in Table 20. The host must not pull nSS low during this period – it must remain high for at least TnSSHR2nSSLR(min) following the SPI transaction that sets PWRDN=1.

To exit low power mode, the host should briefly toggle the nSS SPI line, by taking it low then high again. The time nSS is low must exceed the TnSSL2MISOV(min) timing parameter of Table 20. The host may alternatively perform an SPI transaction, because this pulls nSS low and the CAM312 chip will ignore any data that is sent. For this reason, such an SPI transaction is referred to as a "dummy SPI transaction".

The CAM312 exits low power mode on the falling edge of nSS. It takes some time to restart before it is ready for normal operation including SPI transactions. This time is denoted TPWRDN2nSSL and is specified in Table 34.

Table 34 Power down mode timing

Parameter	Description	Min	Typical	Max
TPWRDN2nSSL	Time between exiting power down and next SPI		1ms	3ms
TnSSLW2nSSLP	Time between exiting power down and SPI transaction(s) used to test the state of MISO for optimised exit from power down	250µs		
TnSSHP2nSSLP	High time for nSS SPI signal required when polling to check whether CAM312 waking from power down	30µs		
TnSSHP2nSSLR	High time for nSS SPI signal required after CAM312 wakes from power down	30µs		

Figure 32 illustrates a process for entering and exiting power down mode. The host starts low power mode by writing a 1 to the SYSCW:PWRDN bit, and stops it by toggling nSS low or a dummy SPI read, as described above. To ensure that the CAM312 has had enough time to start up again after power down mode, the host must allow the maximum TPWRDN2nSSL time specified in Table 34 before the next SPI.

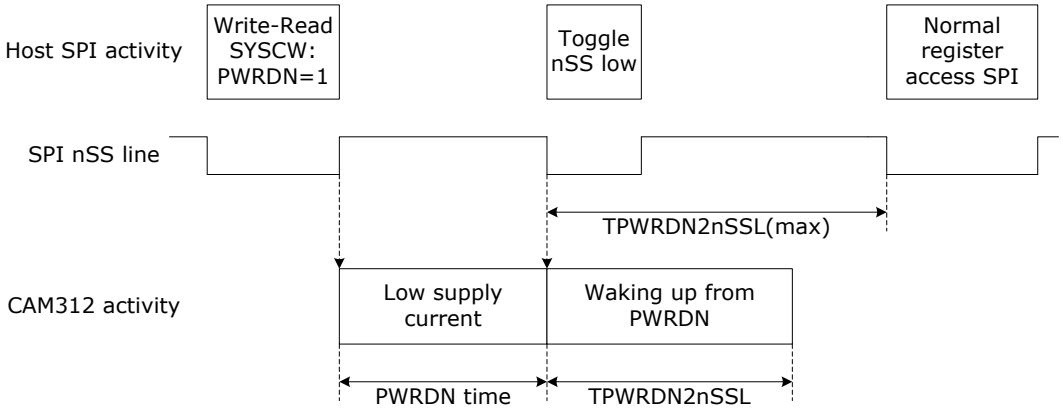


Figure 32 Power down mode, entry and slow exit

Note that the typical TPWRDN2nSSL time specified in Table 34 is much less than the maximum. The CAM312 chip includes a mechanism for signaling that it has finished recovering from power down mode. This allows the host to detect when the CAM312 chip has actually recovered, rather than waiting until the maximum possible time has elapsed. The signaling is done with the CAM312's MISO SPI output line.

In normal operation the MISO line reacts to the host pulling nSS low by changing state from high impedance to the most significant bit of the SYSID word. The SYSID word is 0xABCD by default (section 10.9). The most significant bit in this case is "1" (SYSID bit 15). So, with default SYSID value of 0xABCD, the MISO line normally transitions from high impedance to high following the host pulling nSS low, and before any clocking by the host's SCK SPI output line.

When the CAM312 is still recovering from low power mode, it responds to nSS low by outputting the opposite state on MISO (the inverse of SYSID bit 15). So to detect when the CAM312 is ready for normal operation, the host should pull nSS low and check the state of MISO, without SCK activity. When the state changes (from 0 to 1 if SYSID bit 15 is 1) then the host knows the CAM312 is ready for normal operation again.

Figure 33 illustrates an "optimised exit" process. This is similar to Figure 32, except the host pulls nSS low again after toggling it low to exit power down mode ("hold nSS low").

The time between falling edges on nSS between the toggle and hold is denoted TnSSLW2nSSLP in Figure 33. Its value must exceed TnSSLW2nSSLP(min) specified in Table 34.

Once the CAM312 is ready for normal operation, the state of the MISO line changes. The host should monitor the state of MISO. When it changes to the value of SYSID bit 15, the host should take nSS high again.

nSS must then remain high for a further time period TnSSHP2nSSLR(min) specified in Table 34. Once that period has elapsed, the host may restart normal operation with a register access SPI transaction.

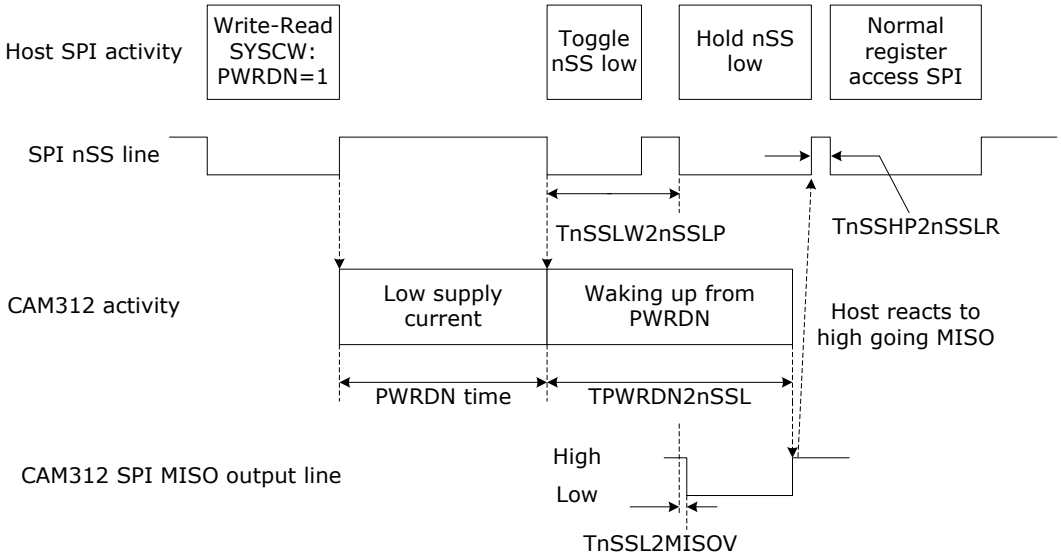


Figure 33 Power down mode, entry and optimised exit (for the case SYSID bit 15 is 1)

The process shown in Figure 33 is illustrated with nSS held low while the host detects the state of MISO. It is also allowable for the host to poll the state of MISO by pulling nSS low periodically while the CAM312 recovers from power down, and checking its state of MISO each time. This allows the host to communicate with other devices on the same SPI bus, during periods when nSS is high. This procedure is illustrated in Figure 34.

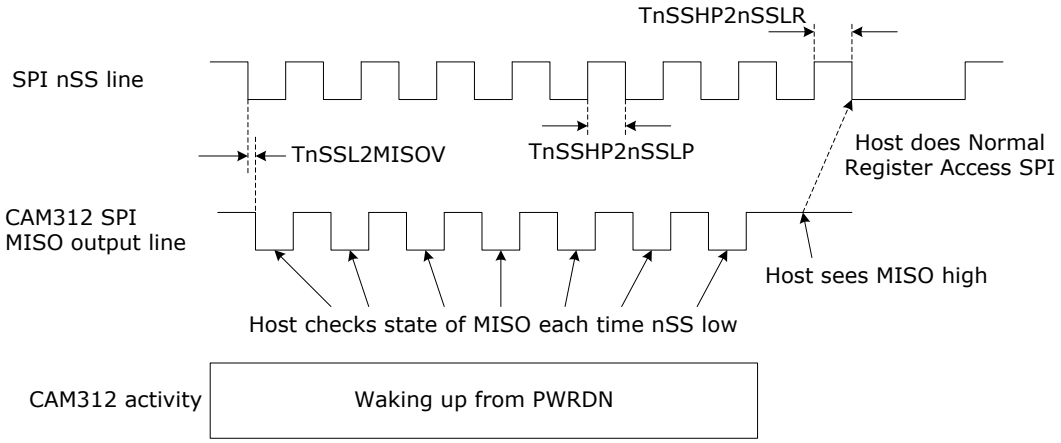


Figure 34 Optimised exit from power down using polling (for the case SYSID bit 15 is 1)

The time between the beginning of the wake SPI transaction and the first nSS low going edge of Figure 34 must exceed TnSSLW2nSSLP(min) specified in Table 34.

The timing parameter TnSSL2MISOV (Table 20) specifies the time it takes for MISO to take its correct state after a low going nSS edge. The host must therefore be sure to sample MISO no earlier than TnSSL2MISOV(max) after each nSS low edge.

Instead of just pulling nSS low each time and checking the state of MISO, the host may perform one-word SPI transactions. The first bit of the CAM312's response on MISO indicates whether the CAM312 is still waking up from power down mode. If this is opposite to SYSID bit 15 then it is still waking up, with typical responses 0x0x005C (if SYSID bit 15 is 1) or 0xF05C (if SYSID bit 15 is 1). Other responses are possible if the CAM312 recovers from power down mode during the SPI transaction. On the other hand, if the CAM312 responds with the SYSID (0xABCD by default) then it has definitely recovered from power down mode.

When polling to check the state of MISO in this way, the host must keep nSS high for a minimum time between lows. This is denoted TnSSHP2nSSLP(min), and is specified in Table 34. This applies whether or not the host also clocks data out of MISO each time nSS is low.

Please note that the CAM312 chip drives its MISO line high when in low power mode. If the MISO line is shared with other devices then the host must not communicate with them over SPI when the CAM312 chip is in low power mode. The CAM312 chip must be taken out of low power mode first. Once this has completed and the CAM312 chip's nSS line is high, the CAM312 chip's MISO output will be high impedance and host communication with the other SPI device may commence.

11.7 Reducing Measurement Supply Current Using Power Down Mode

If there is a period of inactivity between measurements exceeding 3ms, it is possible to reduce supply current by using the CAM312 chip's power down mode in between measurements. Section 12.3 includes examples of the supply current reduction that is possible.

To start the process, the host should first put the CAM312 into power down mode by writing a "1" to the PWRDN bit in the SYSCW register.

Each time a measurement is needed, the host should first wake the CAM312 from power down mode. This is done by toggling nSS low or by sending a dummy SPI transaction, as described in section 11.6.

For lowest average supply current, it is strongly recommended to use an optimised exit from power down mode. This reduces the typical time between coming out of power down mode and the next measurement, which means that the time that the CAM312 spends in power down mode is maximised. To implement optimised exit from power down mode the host should detect the state of MISO while the CAM312 is recovering from power down mode, as described in section 11.6.

Once the CAM312 has woken from power down, the host should follow the procedure for taking an Isolated Single Shot Measurement, described in section 11.5. The SPI transaction labelled "Normal Register Access SPI" in Figure 33 or Figure 34 corresponds to the SPI transaction labelled "Write-Read Sensor 1 SCW" in Figure 31.

Once the measurement results have been read out of the CAM312, the nSS SPI line must be held high for at least TnSSHR2nSSLR(min) as shown in Figure 22 and specified in Table 20. Alternatively, a Checksum Access SPI transaction may be performed as described in section 9.5 to check the validity of the measurement data received.

If the result of the measurement was not VALID, the host must take further Isolated Single Shot Measurements before returning to power-down mode. Measurements must be repeated until either the CAM312 reports VALID, or the number of measurements following wake from power down exceeds the maximum number of measurements on a sensor with in-range target before VALID specified in Table 18. This step is essential, because the resonator frequency search process described in section 8.1 is reinitialised following recovery from power down.

The host should then put the CAM312 into its power down mode again by writing a 1 to the SYSCW word's PWRDN bit.

11.8 Continuous Measurement

Continuous Measurement is for applications where the CAM312 must take repeated measurements on its own, without the host initiating each measurement. A typical application is where the CAM312 is connected to an external DAC and the CAM312 updates the DAC's output each measurement. This is described in section 12.

The host configures the CAM312 for continuous operation in the same way as single shot measurements, as summarised in section 11.2. The host must write a suitable value to the SYSI register to control the Continuous Interval between measurements, see section 10.2.

Next, the host must write to each sensor's SCW register, including GO=CONT=1. If the sensor is Type 2 or Type 6, the host must write an appropriate value for the INCE bit, to either allow or disallow incremental mode. It is recommended to use absolute measurements (INCE=0) to guarantee no loss of absolute position, unless the system needs the fastest possible sample rate and there is no risk of a Fine Skipping Error (section 11.9).

Where only a single sensor is configured for continuous operation, the CAM312 will sample continuously, with an internal timer starting each measurement. If the sensor's sample indicator is mapped to an IO and configured to activate on each new sample and to auto-clear during each measurement (section 10.18), then it will activate as illustrated in Figure 35.

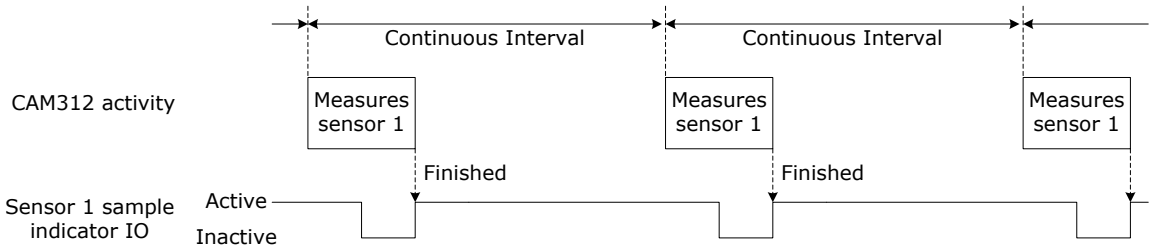


Figure 35 Continuous Measurement, Sensor 1

Where more than one sensor is configured for continuous operation, the CAM312 will measure the sensors in turn, one straight after the other, in order of sensor number. Continuous Interval is measured between successive starts of measurements on the lowest numbered sensor, as illustrated in Figure 36.

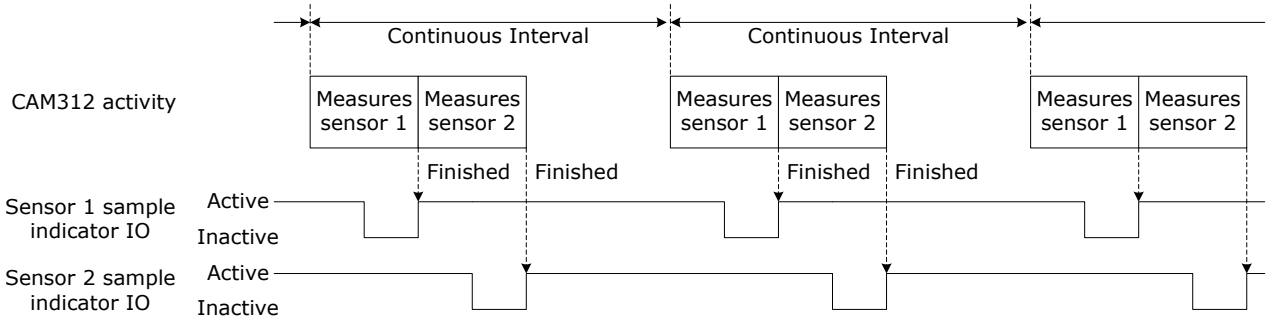


Figure 36 Continuous Measurement, Sensor 1 and Sensor 2

Figure 35 and Figure 36 show how the states of sample indicators change when configured to activate on each new sample (SCTRL=1) and to auto-clear during each measurement (SAUTOCLR=1). To obtain an IO whose state instead reflects whether or not a sensor's latest measurement was VALID, the host should configure sample indicators to activate only when its sensor is VALID (SCTRL=2), and to clear after each measurement (SAUTOCLR=2) if it is not (section 10.18).

The CAM312 can be configured to start continuous measurements after power is applied, without the CAM312 being connected to an SPI interface at the time. This is done using configurable defaults, as detailed in section 0. Register settings are the same as described above. However they are stored in the CAM312 chip's non-volatile memory as configurable defaults. This allows a CAM312 chip to operate autonomously without an SPI connection. A typical application is a product including the CAM312 chip to take measurements connected to a DAC to provide a digital output, as in section 13.

Although not normally recommended, it is possible for a host to read measurement results over SPI during continuous operation. Since the SPI and measurement timings may be asynchronous, the "Measurement in progress" SPI timings of Table 20 must be used.

11.9 Incremental Operation

Type 2 and 6 sensors have two pairs of sensor coils, coarse and fine. Both are patterned for sinusoidal sensitivity. Coarse coils deliver an absolute measurement of position, coarse position, and have relatively poor accuracy and resolution. This is represented by the wobbly upper graph of Figure 37. Fine coils deliver precise fine position, but are only incremental, represented by the middle graph. The CAM312 chip combines the two position measurements so that the position measurements that it reports to the host are always both precise and absolute, as illustrated in the lower graph.

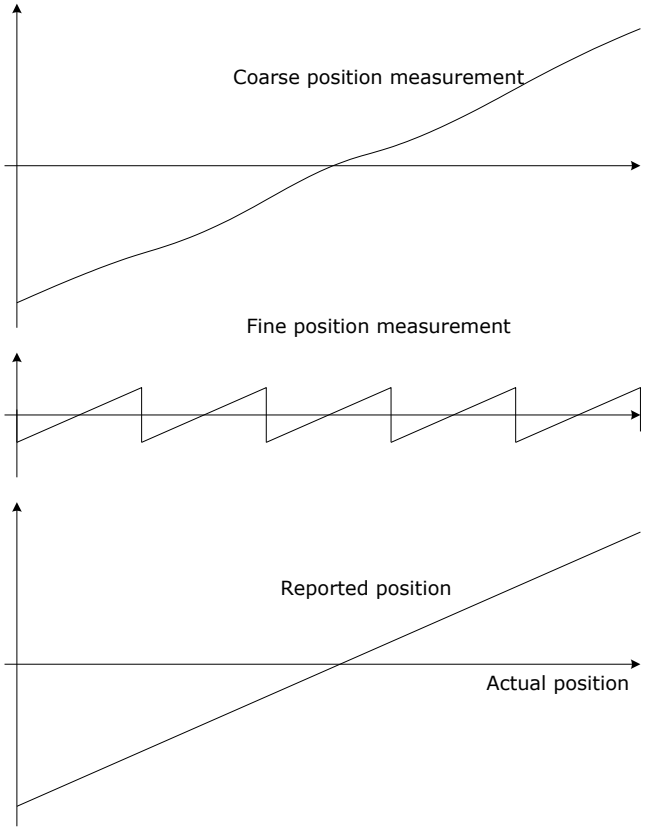


Figure 37 Combining coarse and fine measurements

When measuring such sensors, the CAM312 chip has two modes of operation: Absolute and Incremental. In Absolute mode, each CAM312 measurement includes measurements from both the fine and coarse coils. In Incremental mode, the CAM312 chip uses measurements from fine coils alone. It still reports absolute position, but does so by counting incremental periods from the last absolute measurement. Incremental measurements are only possible after a VALID absolute measurement. If the result of any measurement is NOT VALID, then the system will revert to Absolute mode until VALID once again.

When taking single shot measurements, the host may request Incremental mode measurements by setting the INCE bit (Incremental Enable) to 1. INCE is in the sensor's SCW register, which also contains the INCF bit (Incremental Flag). INCF indicates whether the current result registers contain data from an incremental (INCF=1) or Absolute (INCF=1) measurement.

For lowest supply current and the best available update rate the system should normally be operated in Incremental mode (INCE=1). However in this mode a fine period skipping error may occur if the target moves more than half a fine period per fine measurement. For a Type 6.3 sensor having SinLengthA=120°, the limit is 60°. When operating with a sample interval of 500µs, this means a rotation speed of 16000rpm, allowing for 20% safety margin. For this reason, and as a periodic integrity check, it is recommended to perform an occasional Absolute mode measurement (INCE=0) that will correct any such error. The host may choose when to perform the Absolute mode measurement, for example every 1000 samples or once any time the target's velocity is near zero.

It is strongly recommended to operate in Absolute mode with INCE=0 for operation with an external DAC. There is no chance for a host to request an occasional Absolute measurement, because there is no host connected.

11.10 Measurement Timings

Measurement Time is the time between the host requesting a measurement over SPI and its results being available for reading over SPI. It is measured from the rising edge of the nSS signal of the SPI transaction setting a sensor's GO bit and the activation of an associated sample indicator. Please see Figure 28.

Measurement Time depends on the Type of sensor connected and whether or not a measurement is VALID. For Type 2 and 6 sensors, the host also has the option to request incremental operation, by setting INCE=0. If the CAM312 is able to perform an incremental measurement it sets INCF, resulting in a shorter Measurement Time.

If VALID, measurement time also depends on the frequency of the resonator inside the target. The maximum measurement time is obtained when resonator frequency is at its lowest limit. This is when the reported Relative Frequency is approximately -25000Hz.

Table 35 specifies the Maximum Measurement Time as a function of Sensor Type, VALID and INCF. If the host performs an SPI transaction during a measurement the Measurement Time will increase and may exceed the value of Table 35.

Sample Interval is the time between successive measurements. When operating in repeated single shot mode, it is the time between successive SPI transactions used to start and read measurement results, as illustrated in Figure 38.

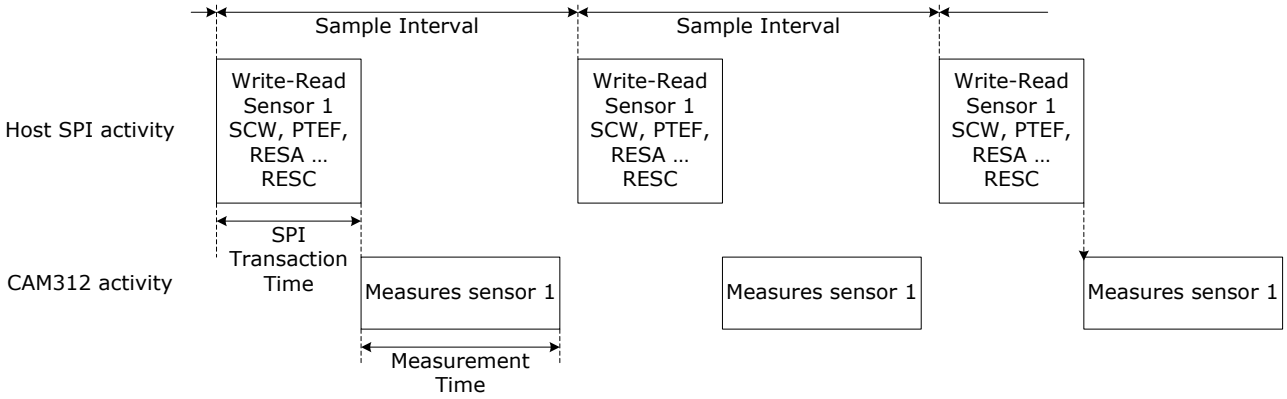


Figure 38 Definition of Sample Interval

Achievable Minimum SPI Transaction Times are detailed in section 9.7. In Table 35, the Minimum Sample Interval is equal to these figures plus the Measurement Time. Maximum Sample Rate is the reciprocal of Minimum Sample Interval.

Group Delay is defined as the time between the effective "capture" point of position and the activation of a sample indicator at the end of measurement. Smaller values are obtained with Type 4 sensors, or Type 2 or 6 operating in Incremental Mode (INCF=1). Group Delay is also listed in Table 35. Other sources of system delay include the time it takes a host to respond to the sample indicator, and the SPI Transaction Time noted above for the host to read out measurement data over SPI.

Table 35 Measurement Time and Group Delay

Sensor Type	VALID	INCF	Maximum Measurement Time	Minimum Sample Interval	Maximum Sample Rate	Group Delay
4	0	N/A	430µs	444µs	2250Hz	-
	1	N/A				200µs
2 or 6	0	N/A	647µs	667µs	1500Hz	-
	1	0				340µs
	1	1	440µs	460µs	2170Hz	200µs
3	N/A	N/A	2180µs	2200µs	454Hz	1500µs

11.11 Configurable Defaults

The CAM312 chip's default register contents can be configured over the SPI interface ("Configurable Defaults"). To change to new Configurable Defaults, the host should first write the required settings to the CAM312 chip's registers. The host should then write 0x0C1C to the SAVEKEY register (section 10.6) followed by a 1 to the SAVE bit (section 10.1). The CAM312 will then reset itself. After this reset, and each subsequent one, the CAM312 operation is determined by the saved Configurable Defaults.

The CAM312 chip's Factory Defaults can be restored by writing 0x0C1C to the SAVEKEY register followed by a 1 to the FACTORY bit (section 10.1). The CAM312 will reset itself following this operation.

The Configurable Defaults are stored in FLASH memory. The number of FLASH updates are limited (section 2.5), so updates to Configurable Defaults should be rare. The SAVE and FACTORY operations each count as one FLASH update. For most applications the update will be done only once when a product containing the CAM312 is manufactured.

The most common application for Configurable Defaults is to generate an analog output whose value depends on position. In this case there is no host SPI interface connection during normal operation. An SPI interface is connected temporarily during manufacturing of the customer's product, to program the Configurable Defaults. Configurable Default values will include operation in continuous mode (section 0), and settings required for the CAM312 to drive an external DAC (section 12).

It is strongly recommended NOT to use Configurable Defaults to store the configuration settings described in section 11.2, when measurements are to be read by a host over SPI. It is recommended to store the configuration settings within the host, and to write them afresh to the CAM312 after any reset. This avoids any possible discrepancy between what the host "thinks" the configuration settings should be and what is already preprogrammed into the CAM312. This discrepancy could in principle be avoided by the host reading all of the CAM312's register settings after a reset. However if the host were programmed to read and check settings, it might as well program them afresh instead. The exception is the SYSID register in cases when there are multiple CAM312 devices sharing the same SPI bus. In this case each device may have its own unique SYSID value, so that the host can check any data it receives comes from the expected device. The SYSID values would be programmed during manufacturing of the customer's product. Please refer to section 10.9 for more on the SYSID register.

Configurable Defaults revert to factory settings after an update of CAM312 Application Code (section 14). If it is important that custom settings survive an update, they should be read out of the CTU before the update process and written back and saved afterwards.

11.12 Status Codes

In normal operation the first word of each SPI transaction begins with the CAM312 chip's System ID, SYSID (section 10.9). Other codes may be encountered instead, and their meanings are described in Table 36.

Table 36 Status Codes

First word of SPI transaction	Status description
SYSID (0xABCD by default)	Normal operation
0x10AD	CAM312 chip is in Bootloader mode expecting Application Code over the SPI interface
0x0BAD	CAM312 chip is in Bootloader mode and has detected SPI transactions that do not include valid Application Code
0x600D	Bootloader data block transfer has been successful
0x005C	Waking up from power down, when SYSID bit 15 = 1 (includes default value of SYSID of 0xABCD), see section 11.6.
0xF05C	Waking up from power down, when SYSID bit 15 = 0, see section 11.6.
Other words starting 0x1 and 0xE	Reserved

If Status Codes relating to Bootloader mode are seen unexpectedly, they are usually due to SPI interface activity too soon after a reset (section 11.1). If they do not appear to be related to an intended reset, it is likely that an unintended reset occurred, for example due to an unstable power supply voltage or unexpected activity on the CAM312 chip's nRESET pin.

12 Supply Current

This section provides values for supply current. This is the total current drawn by the CAM312 chip and its external circuitry from VSUPPLY. For example, when connected to one or more Type 4 sensors, the sum of the current passing through R_VDD and R_EX shown in Figure 7. Unless otherwise noted, values are for typical components operating at room temperature with a target present and reporting VALID.

12.1 No Measurement in Progress

Table 37 shows typical operating current for the system when there is no SPI activity or sampling in progress. Supply current depends on whether or not the host has put the CAM312 into low power mode using the PWRDN bit. For details of how to put the CAM312 into its low power mode, and taking it out again, please see section 11.6.

Table 37 Typical current drawn from VSUPPLY, no sampling in progress

Sensor circuitry	PWRDN=0	PWRDN=1
Type 2	8mA	200µA
Types 2, 3, 4, 6	8mA	12µA

Type 2 circuitry draws considerably more current when PWRDN=1. This is because its bias network (R_BH, R_BL of Figure 10) is connected. For applications requiring lower supply current with a Type 2 sensor connected, the host may apply a switched 3.3V supply to R_BH, for example from an IO. Drive the IO high for normal operation, and low or high impedance for low power.

12.2 Measurement in Progress, High Sample Rate

Table 38 shows typical current consumption figures for the system when sampling is in progress. These are measured from the VSUPPLY line, which supplies both the CAM312 chip and the excitation circuitry. It is therefore the total of all current needed to supply the sensor system.

Average current consumption depends on sample rate. For Type 2 and 6 sensors, it also depends on whether the CAM312 is operating in incremental or absolute modes (section 11.9).

Table 38 includes both average and peak current values. Peak current occurs while excitation waveform is being generated (Figure 5) to drive current into the excitation coil.

Table 38 Typical current drawn from VSUPPLY, operating at high sample rate

Sensor	Samples / Second	Average Current	Peak Current (C_EXSUP=10µF)	Peak Current (C_EXSUP=100µF)
Type 4 (100mm Type 1 Linear Sensor)	2000	25mA	80mA	45mA
Type 2 (350mm Type 2.12 Linear Sensor)	1000 (INCE=0)	40mA	100mA	60mA
	2000 (INCE=1)	39mA		
Type 6 (35mm Type 6.3 Rotary Sensor)	1000 (INCE=0)	25mA	80mA	45mA
	2000 (INCE=1)	24mA		
Type 3 (78mm Type 3.2 Linear Sensor)	450	53mA	80mA	45mA

Peak current depends on the value of the excitation decoupling capacitor C_EXSUP. The minimum recommended value is 10µF, as specified in sections 3.3, 4.2 and 5.2. A higher value may be used. Table 38 includes peak current values for both 10µF and 100µF to illustrate the difference.

The values for peak current shown in Table 37 are largely independent of sample rate. They are based on component values shown for the relevant sensor Type, for example Table 9 for Type 4. If an application only requires a relatively low sample rate, for example 100 samples per second, the peak current may be reduced further by increasing the value of R_EX. The value should be chosen so that the peak voltage drop across R_EX is no more than 0.3V.

Care must be taken to ensure the VSUPPLY line's voltage remains above the minimum Operating Supply Voltage specified in Table 3 at all times, including when peak current is being drawn.

12.3 Measurement in Progress, Low Sample Rate

This section provides examples of average supply current when taking measurements at a relatively low rate. Please refer to section 12.1 above for supply current with no sampling, and section 12.2 for supply current when sampling near maximum rate.

Table 39 includes figures for typical supply current at low sample rate. This depends on what sensor Type is being processed and the sample rate. It also depends on whether the host puts the CAM312 chip into power down mode between measurements. "Without PWRDN" figures are based on the measurement process described in section 11.3. "With PWRDN" figures are based on the low-power measurement process described in section 11.7 and SPI timings near minimum.

Table 39 Typical average current drawn from VSUPPLY, operating at low sample rate

Sensor	**Samples / Second**	**Average Current**	
		Without PWRDN	**With PWRDN**
Type 4 (100mm Type 1 Linear Sensor)	200	9.6mA	5.3mA
	50	7.8mA	2.3mA
Type 2 (350mm Type 2.12 Linear Sensor, INCE=0)	200	12.7mA	8.3mA
	50	8.8mA	3.2mA
Type 6 (35mm Type 6.3 Rotary Sensor, INCE=0)	200	10.9mA	6.9mA
	50	8.2mA	2.7mA
Type 3 (78mm Type 3.2 Linear Sensor)	50	12mA	7.3mA

Important note for Type 2 sensor applications: as noted in section 5.1, the schematic shown in that section requires modification for taking repeated measurements using low power mode ("with PWRDN"). Please contact CambridgeIC for alternatives.

13 Analog Output from an External DAC

13.1 DAC Overview

The CAM312 CTU chip has an SPI interface which can be used to read the results of sensor measurements. This interface yields the best performance, due to its digital nature, and is inherently low-cost. However in some applications, such as those lacking a digital processor or requiring support for a legacy interface, analog output may be required. To support these requirements, the CTU is able to drive an external Digital to Analog Converter (DAC).

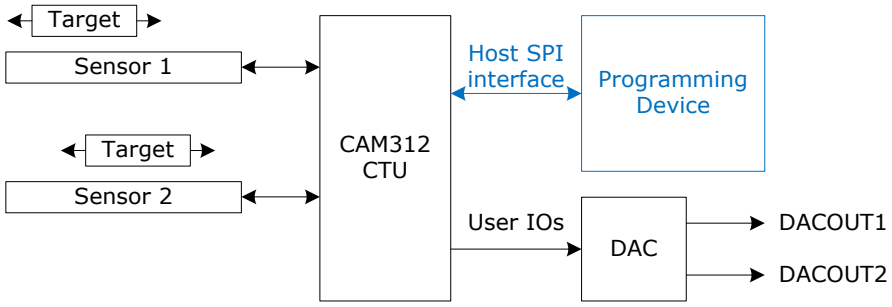


Figure 39 CAM312 CTU chip driving and external DAC

The CAM312 CTU chip drives external DACs from its user IO pins, as illustrated in the block diagram of Figure 39. The CAM312 supports up to 2 Type 4 sensors, and can drive DACs with up to 2 outputs.

When driving an external DAC, the CAM312 is usually operated stand-alone, without an external host device connected. This requires the chip's configuration to be programmed into non-volatile memory over the SPI interface (section 11.11). Programming is usually performed when the product containing the CAM312 chip is manufactured. Programming may be performed by a device that implements the SPI interface and register protocols described in this datasheet. Alternatively, a CTU Adapter from CambridgeIC may be used to configure the chip from a PC running CambridgeIC CTU Software.

13.2 Circuit Connections between CTU and DAC

The CAM312 CTU drives an external DAC with an SPI interface. The CTU is the SPI master device. When configured for DAC operation, IO2 is the serial clock, IO3 is the serial slave select/ synchronise and IO4 is the serial data. Figure 40 shows how these signals are connected to a DAC having typical SPI interface names.

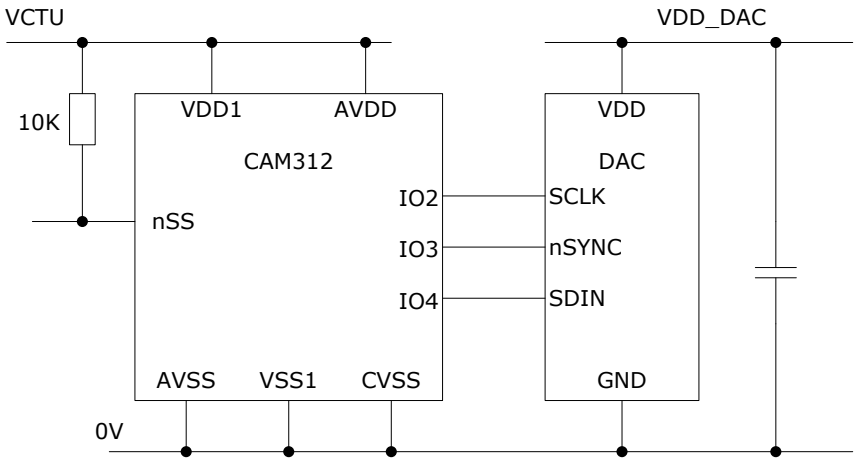


Figure 40 connecting the CTU to an external DAC

The DAC's power supply voltage ("VDD_DAC") need not equal that of the CAM312 chip ("VSUPPLY") providing the CAM312 chip's IO output voltage levels (section 2.4) are compatible with the DAC's logic inputs.

The CAM312 chip's nSS SPI input must be kept high when the CAM312 is operating in continuous mode and outputting data to an external DAC. It must not be left floating, because a low level on nSS will be interpreted as an SPI transaction that could result in unexpected behavior. In the absence of any other circuitry to ensure nSS remains high, a pull-up resistor should be used as illustrated in Figure 40.

13.3 Supported DACs and Formats

The CAM312 CTU supports DACs having SPI interfaces with the message formats listed in Table 40. The DACFORMAT[2:0] bits within the SYSDAC register (section 10.4) select which format the CTU uses. Bit number 15 is the first one clocked into the DAC, and bit number 0 the last. The SPI clock rate specification is in Table 44.

When DACFORMAT[2:0]=0, each SPI message includes the two least significant bits of the DACADDR[2:0] bits from a sensor's DACCW register (section 10.16). These enable each sensor to address a different DAC output when the CTU is connected to an appropriate DAC. The function of the DACOP[1:0] bits is specific to each DAC family. DACOP[1:0] settings usually include power down, and provision for synchronously updating DACs in multi-output systems.

DACFORMAT[2:0]=1 is similar, except it excludes the DACADDR bits and is therefore only used for DACs with a single output.

In both cases, DACDATA is the data controlling the DAC output voltage. Its value depends on sensor position measurements and whether or not the last measurement was VALID. Please see section 13.4 for an overview of how DACDATA is generated, and the following sections for more detail. The CTU calculates DACDATA to 16-bit precision internally. The number of bits sent over the SPI interface depends on DACFORMAT[2:0].

DAC families often include parts with different resolutions, typically from 8 to 12 bits as illustrated in Table 41. The same CTU configuration is valid for each part, since DACs with lower resolutions ignore the state of the appropriate number of least significant bits.

Table 40 SPI message format

DACFORMAT [2:0]	SPI Data Bits in SPI message from CTU to DAC															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	DACADDR[1:0]		DACOP[1:0]		DACDATA[15:4]											
1	DACOP[1:0]		DACDATA[15:2]													

Table 41 Examples of supported DACs

Application	Outputs per DAC	DAC Resolution			DAC FORMAT [2:0]	DACOP[2:0] normal operation	SPI Mode
		8 bit	10 bit	12 bit			
Low-cost	1	DAC081S101	DAC101S101	DAC121S101	0	0	0
	2	DAC082S085	DAC102S085	DAC122S085	0	1	0
Precision	1			AD5621B	1	0	0

13.4 DAC Code Generation Overview

This section is an overview of how the CAM312 chip generates the SPI message to send to an external DAC. Figure 41 illustrates the processing steps involved. Calculation steps are detailed in the following sections.

Processing begins with calculation of reported position, including whether or not the measurement was VALID. If VALID, the CAM312 chip makes reported position data available over the normal slave SPI interface, to be read by a host device if present.

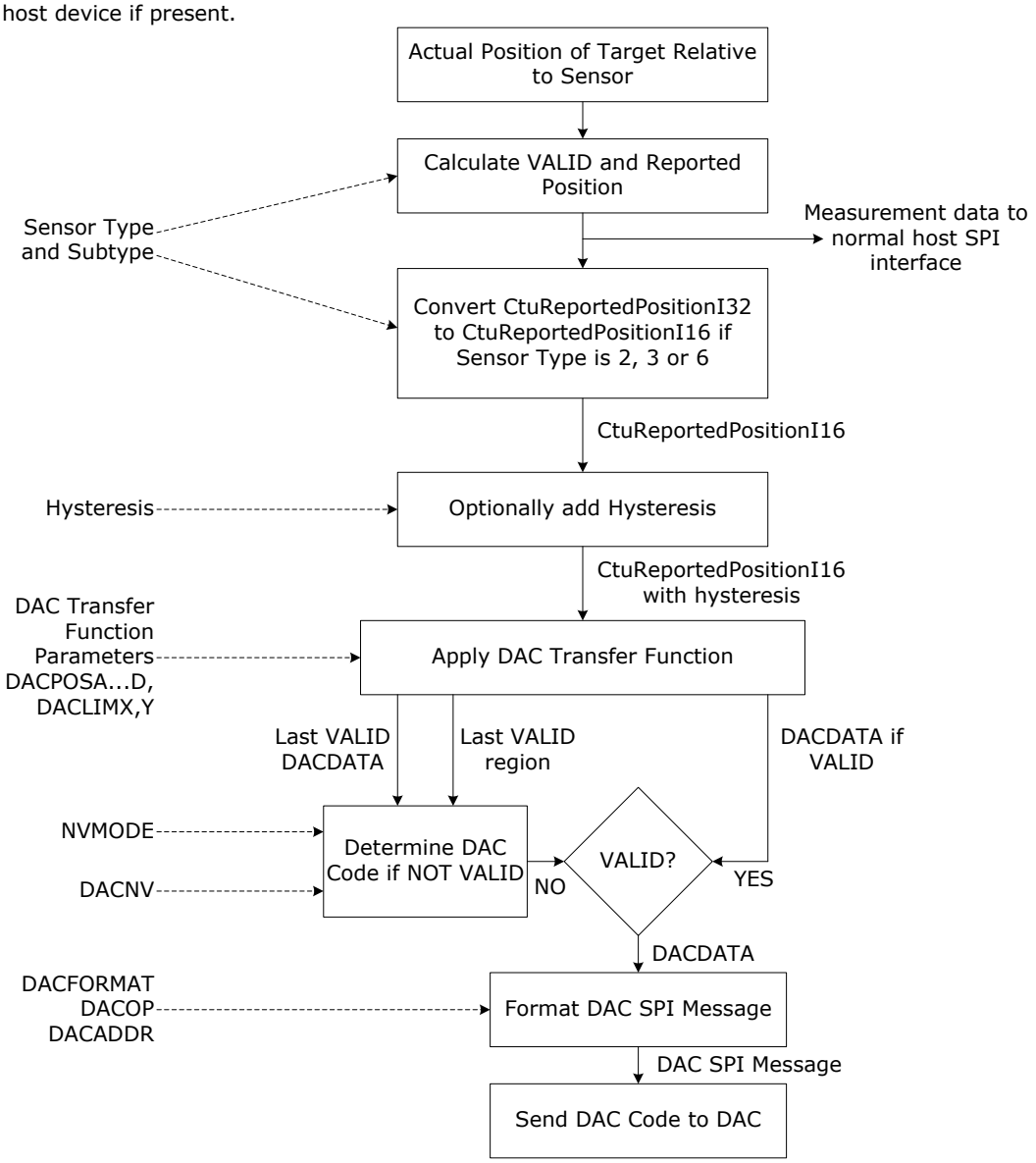


Figure 41 DAC Code Generation Process

The next step is conversion of reported position data from 32-bit to 16-bit format. This is only necessary for Sensor Types 2, 3 and 6, because their format for reporting position over the host SPI interface is 32-bit. Please see section 13.5.

The CAM312 chip's measuring process results in position data that is completely free from any unwanted hysteresis. However in some cases it may be desirable to deliberately add hysteresis to position data in order to prevent code jumps in DAC data caused by random noise. The CAM312 chip includes an option to add hysteresis for DAC output, see 13.6.

The CAM312 then applies a "DAC Transfer Function" which allows the DAC code to be scaled, offset, limited and reversed as required for the end application. This process yields a 16-bit number DACDATA which is included in the SPI data sent to the DAC when measurement data was VALID. DACDATA[15:0] is the code representing DAC output,

with 0x0000 (decimal 0) typically representing the minimum output voltage and 0xFFFF (decimal 65535) representing the maximum. Please see section 13.7.

The CAM312 can be configured to behave in different ways in the case that measurement data was not VALID, under control of the DACNV parameter. Please see section 13.8.

Finally, the CAM312 formats the SPI message to be sent to the DAC according to the chosen DACFORMAT setting, as defined in Table 40.

13.5 Scaling 32-bit to 16-bit Reported Position

The processes for calculating the data to be sent to the DAC are done on a 16-bit position values. When measuring Type 4 sensors, the CAM312 returns a 16-bit result. However, when measuring Type 2, 3 or 6 sensors, the CAM312 delivers a 32-bit result. In this case, the conversion is described by Equation 5…

$$CtuReportedPositionI16 = \frac{CtuReportedPositionI32}{SubType} = \frac{Position\ in\ Measurement\ Units}{SinLength \times Subtype} \times 65536$$

Equation 5 Conversion from 32-bit to 16-bit representations of reported position

The sensor's Subtype is the number of fine cos/sin periods per coarse cos/sin period, and is defined in its datasheet. Its SinLength value is the fine pitch in measurement units, also defined in the sensor datasheet.

For absolute 360° sensors, SinLength x Subtype = 360°, so Equation 5 yields CtuReportedPositionI16 in the range -32768 to +32768 representing the angle range -180° to +180°.

For absolute linear sensors, SinLength x Subtype is always somewhat greater than the sensor's Measuring Range. This means that the resulting values of CtuReportedPositionI16 span somewhat less than the full range -32768 to +32768.

13.6 Optional Hysteresis

In some applications it may be desirable to deliberately add hysteresis to position data in order to prevent DAC code jumps in DAC data caused by random noise. In most applications hysteresis is unwanted and inappropriate because it results in apparent "stickiness" in the output which introduces errors and can upset certain control systems. However the facility is available if it is needed.

Figure 42 illustrates a mechanical analog of the hysteresis algorithm together with the simple calculations involved. The parameter "h" is the amount of hysteresis required, and its value is the HYSTERESIS bit field in the CAM312's PTAH sensor register (section 10.19). In(n) is the new input ("nth input"), out(n-1) is the previous output ("n-1 th output) and out(n) is the new output.

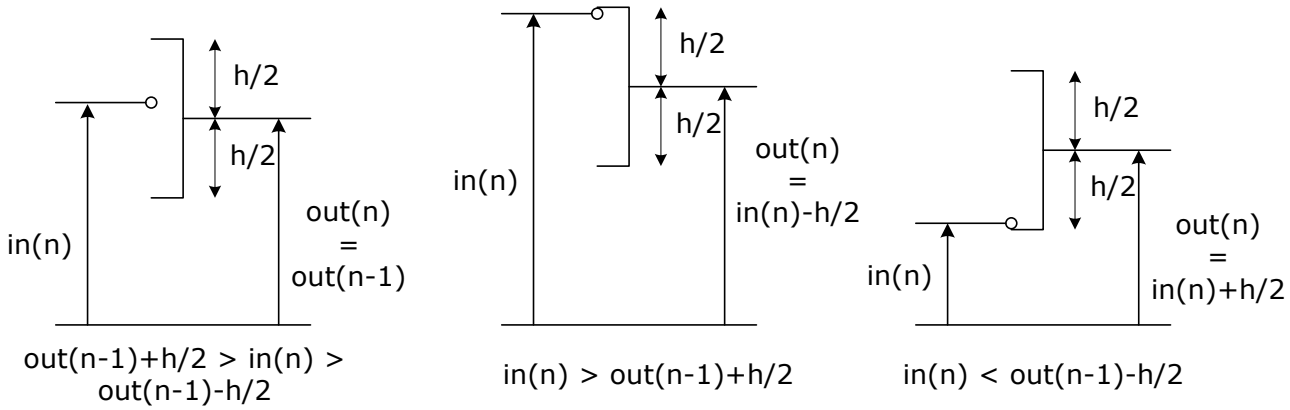


Figure 42 Illustration of function for adding hysteresis

The CAM312 performs the hysteresis calculation using modulo arithmetic so that the function behaves smoothly across the apparent discontinuity between input and out values -32768 and +32767.

For normal operation free from hysteresis keep HYSTERESIS set to 0 (its default value).

13.7 DAC Transfer Function

The DAC Transfer Function is the relationship between measured position (after hysteresis) and the DACCODE[15:0] value used to control DAC output level. It is configured using the parameters listed in Table 42.

Table 42 parameters defining the DAC Transfer Function

Parameter name	Name in Figure 43	Summary of parameter's function	Register description
DACPOSA	A	Position at start of up slope	Section 10.14
DACPOSB	B	Position at end of up slope	
DACPOSC	C	Position at start of down slope	
DACPOSD	D	Position at end of down slope	
DACLIMX	X	Minimum DAC value when valid	Section 10.15
DACLIMY	Y	Maximum DAC value when valid, must be greater than DACX	

Figure 43 illustrates the range of DAC Transfer Functions available. Figure 43(a) is a key to the other graphs. The x-axis is CtuReportedPositionI16 after hysteresis is applied, running from -32768 to +32767. The y-axis is the resulting DAC value.

Figure 43(b) to Figure 43(d) illustrate the range of possible Transfer Functions applicable to linear sensors. The shape of the Transfer Function is configured using the programmable parameters listed in Table 42. DACPOSA-F define the start and end positions of the sloping portions of the Transfer Function. DACX and DACY define minimum and maximum DAC values when the sensor's measurement is valid.

Figure 43(b) is the *general case* from which all other Transfer Functions are derived. The DAC output increases linearly between DACX and DACY for positions between DACPOSA and DACPOSB. It remains at DACY between DACPOSB and DACPOSC. It decreases linearly from DACY to DACX for positions between DACPOSC and DACPOSD. It remains at DACX from DACPOSD upwards (back to DACPOSA in the case of a rotary sensor).

Figure 43(c) illustrates a common application where the DAC output increases linearly between DACPOSA and DACPOSB, typically at the ends of the sensor's measuring length. When position is valid but less than DACPOSA the DAC output remains at DACX, and when it is valid but greater than DACPOSB the DAC output remains at DACY. DACPOSD should be set to -32768 and DACPOSC to +32767.

Figure 43(d) is similar to (c), except that the DAC output slopes in the opposite direction. This can only be achieved as shown, with DACPOSA-D repositioned, and not by selecting a value of DACY smaller than DACX.

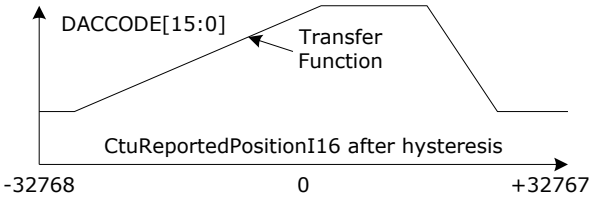
The Transfer Function of Figure 43(c) arises when DACPOSA-D are re-ordered from (b).

Rotary sensors behave in the same way, except that *modulo arithmetic* applies. Positions -32768 and +32767 are adjacent. The transfer functions can be *rotated* to start at any angle offset, as illustrated in Figure 43(f).

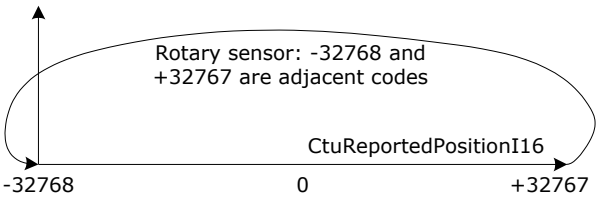
Figure 43(g) illustrates how to achieve a DAC output which increases linearly with rotation angle. The start angle of the slope is defined by DACPOSA = DACPOSD. The end angle is defined by DACPOSB = DACPOSC, which should be set to 1 less than DACPOSA and DACPOSD. (h) is similar, except with sloping in the opposite direction.
Figure 43 (i) and Figure 43(j) are more general cases applicable to rotary sensors, and are rotated versions of (b).

DACPOSA…D must be correctly ordered for the CTU to generate meaningful data for the DAC. The general rule is illustrated in Figure 43(k). Position is treated as an angle, so that the values +32767 and -32768 represent adjacent positions. DACPOSA must be anticlockwise from or equal to DACPOSD. DACPOSB must be anticlockwise from DACPOSA. DACPOSC must be anticlockwise from or equal to DACPOSB. Finally DACPOSD must be anticlockwise from DACPOSC.

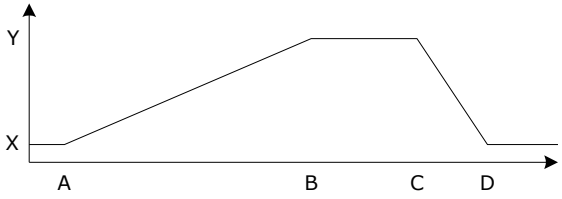
DACLIMY must be greater than or equal to DACLIMX. If a "flipped" transfer function is required, the values of DACPOSA and DACPOSC must be swapped, and those of DACPOSB and DACPOSD.



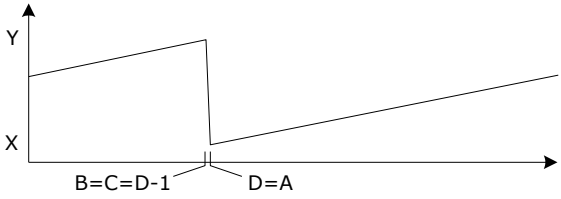
(a) key to other graphs



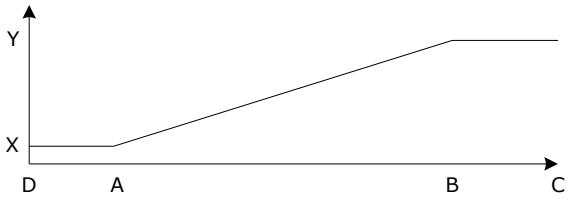
(f) rotary sensor behaviour



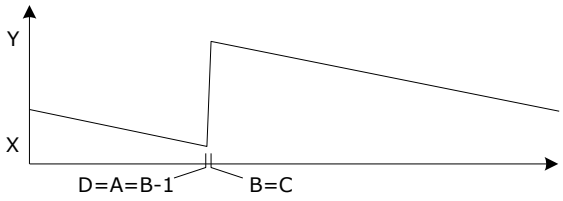
(b) positive then negative slope



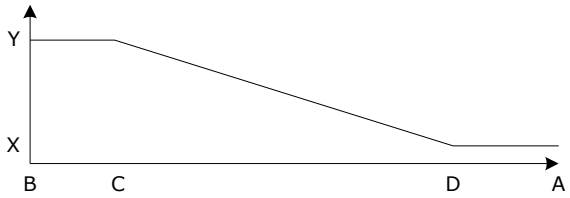
(g) positive slope with configurable position offset



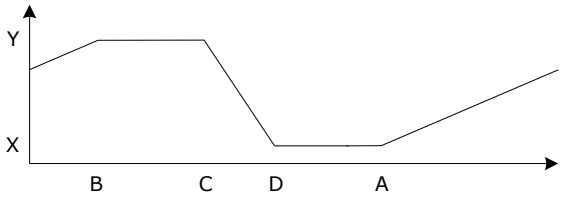
(c) positive slope with dwell at ends



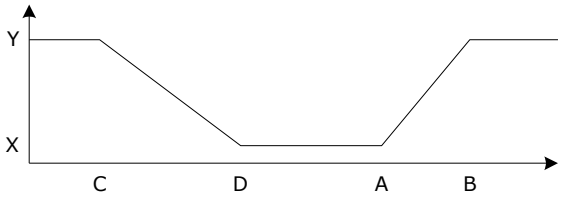
(h) negative slope with configurable position offset



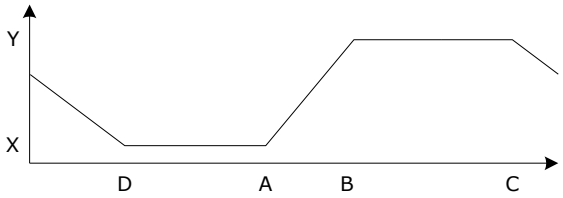
(d) negative slope with dwell at ends



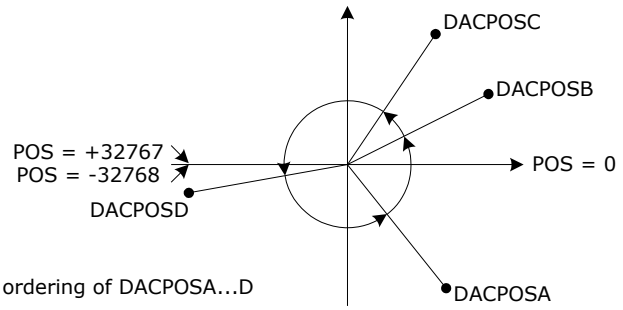
(i) positive slope spans phase jump



(e) negative then positive slope



(j) negative slope spans phase jump



(k) guide to ordering of DACPOSA...D

Figure 43 configuring the DAC Transfer Function

13.8 DAC Output When Measurements are Not Valid

When the CAM312 chip's DAC output is on and a sensor's DAC output is enabled, the CAM312 sends an SPI message to the connected DAC after each measurement including suitably formatted DACDATA. When the measurement was VALID, DACDATA is the output of the transfer function detailed in section 13.7. This section describes the options available in the case that the measurement was not VALID.

The behavior when not VALID is configured with the NVMODE bits in the SYSDAC register (section 10.4). There are 3 possible modes, defined in Table 43.

Table 43 DAC output on invalid measurements

NVMODE	Mode Name	DACDATA when last sensor measurement was invalid
0	Default Value Mode	Forced to DACNV
1	Last Value Mode	Holds value from last measurement. Defaults to DACNV if there was no valid measurement since the last reset.
2	End Latch Mode	If last valid measurement was in region DA, outputs DACX. If last measurement was in region BC, outputs DACY. Otherwise outputs DACNV.
3	Reserved	

In Default Value Mode, the DAC is driven to the DACNV value stored in the sensor's DACCW register (section 10.16). DACNV[15:8] is an 8-bit wide field representing the upper 8 bits of DACCODE[15:0]. The CTU sets the lower 8 bits to 0. The value of DACNV may be selected such that the DAC output falls significantly outside the range DACLIMX…Y. That way, a device connected to the analog output alone can distinguish between a valid and invalid measurement.

In Last Value Mode, the DAC is driven with the value of DACDATA from the last VALID measurement when available. This may suit some systems that can experience a loss of target signal, e.g. due to momentarily increase gap, and where the DAC output should still reflect an estimate of position.

End Latch Mode is similar to Last Value Mode, except that the last position only "latches" when the last value fell in regions DA or BC. This behaviour is useful for linear sensors where the target is allowed to move beyond the ends of the sensor's measuring length. With the Transfer Function configured as Figure 43(c) or (d), the DAC outputs either DACLIMX or DACLIMY depending on the direction the target left the sensor. If the last measured position was in region AB or CD, the DAC is driven with DACCODE = DACNV to indicate that the target did not leave the sensor in one of the expected locations.

Note that the first few measurements from a sensor after reset may not be VALID due to the frequency search process that the CTU undertakes (section 8). Appropriate DACNV and NVMODE settings should therefore always be considered.

13.9 DAC Calibration

In some applications it may be possible to establish DAC configuration in advance, so that all products are programmed with the same CAM312 CTU chip configuration.

Alternatively, DAC configuration may be established as part of the manufacturing and test process of the product. For example, there may be mechanical tolerances requiring the optimum values of DACPOSA…D to be established, and/or the need to calibrate out errors in the DAC output and its signal conditioning.

Figure 44 illustrates a typical CTU-based system undergoing calibration. A target positioning system positions the target at pre-selected positions, for example at the required ends of the measuring region. The CTU's reported position at these locations may be measured over the SPI interface and averaged by the customer's Test and Configuration System, ready for programming back into the CTU's DACPOSA…D registers as appropriate.

To calibrate out errors in the DAC and any signal conditioning, the Test and Configuration System may need to set the value of DACCODE controlling the DAC output to a specific value. The DACCAL bit in the SYSDAC register (section 10.4) is available for this purpose. When set, the CTU sends an SPI message to the DAC with the DACCODE value equal to the required sensor's DACLIMX, DACLIMY or DACNV each time one of those registers is written. DAC configuration should otherwise be as detailed in section 13.12.

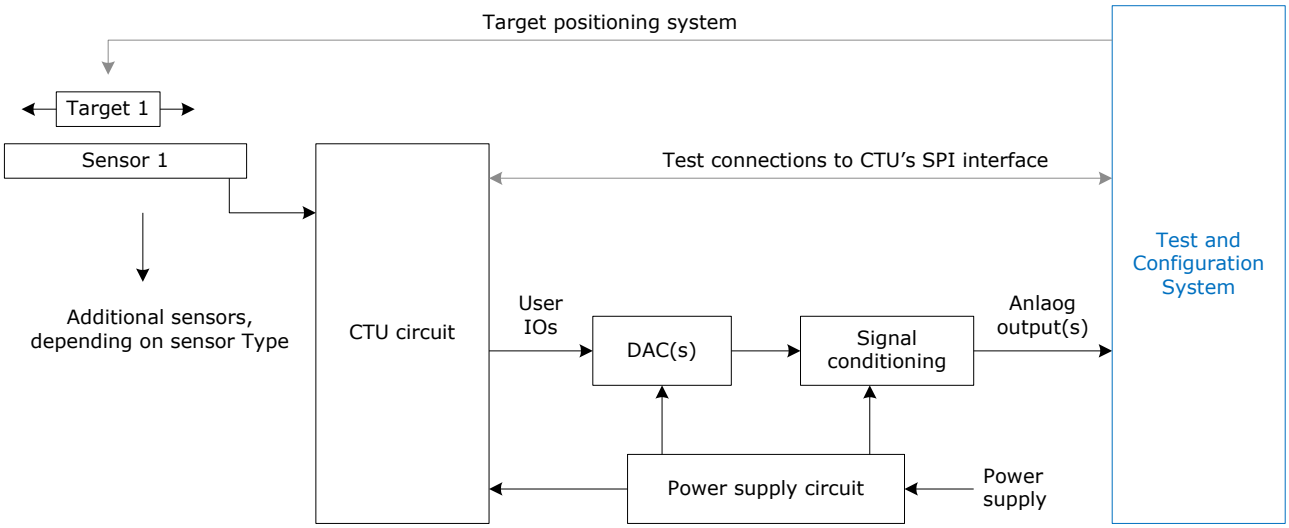


Figure 44 block diagram of a CTU-based sensor system undergoing calibration

13.10 Synchronisation of DAC Outputs

When the CTU is connected to more than one sensor it measures each in sequence. If a multi-output DAC is connected, the CTU updates the DAC register associated with each sensor when each individual measurement completes. With the multi-output DACs of Table 41 and the value for DACOP[1:0] shown, each DAC output will be updated when its register is updated, with the updates staggered in time accordingly.

Some systems may require that changes in the output voltages of a multi-output DAC occur at the same time. The multi-output DACs shown in Table 41 allow synchronisation. DACOP[1:0] should be set to 1 for one "trigger" sensor and 0 for the remaining "follower" sensors. When the CTU updates DAC register values for the "follower" sensors the analog outputs will not change until the "trigger" sensor is measured and its DAC register is updated. That way, all DAC outputs will be updated synchronously.

13.11 LED Control

When the CAM312 CTU chip is connected to a host system over the normal host SPI interface, it provides system health monitoring information including Amplitude and Relative Frequency data (section 10.12). The host can use this information to monitor system health. For example it can provide a warning if a reduction in Amplitude is observed, which may be caused by the mechanical gap between target and sensor becoming excessive.

When the CAM312 chip is connected only to an external DAC providing an analog output and there is no host SPI connection, it can be helpful to have an indication of system health, however basic. To this end, the CAM312 can be configured to control an LED to provide a basic visual indication of system health.

Figure 45 shows how an LED may be controlled by one of the CAM312 CTU chip's user IOs using a MOSFET and current limiting resistor. The IO is configured for signaling a sensor's status with LED(s) by configuring the sensor's LEDCW appropriately (section 10.13). Each sensor can be mapped to any user IO with the LEDMAP[3:0] bits. When operating with an external DAC connected, IO2, IO3 and IO4 are used for the SPI interface for the DAC. This leaves IO1, IO5 and IO6 available for LED control.

The IO will flash the LED at approximately 2Hz when the sensor's last measurement was VALID and Amplitude was less than LEDTHRESHOLD[11:0]. This is an indication that the target is in range, but Amplitude has not yet reached a desired minimum value to yield the performance required.

The IO will light the LED continuously when the sensor's last measured Amplitude was greater or equal to LEDTHRESHOLD[11:0] to indicate a healthy signal level.

LED control operates independently of host SPI transactions when the CTU is operating in continuous mode (section 11.8).

IOs used to drive LEDs are configured for active high or active low using the SYSIO register (section 10.3). In the example shown in Figure 52, the IO should be configured for active high. For changes to the SYSIO register to apply to an IO controlled LED, they must be made with LEDMAP[3:0] set to 0.

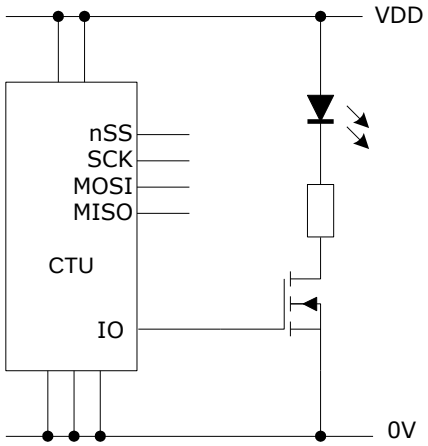


Figure 45 Driving an LED

13.12 Summary of Register Settings for DAC Operation

This sub-section summarises the register settings typically required for autonomous CAM312 CTU chip operation with an external DAC, with links to sections to learn more.

Register(:Bit)	Function Controlled	Section Ref
SYSI	Interval between measurements. Set to 0 for fastest update rate, or greater values for lower power operation.	10.2
SYSDAC:DACSON	Set to 1	10.4
SYSDAC:CPOL,CPHA	Set to match the selected DAC's SPI Mode	
SYSDAC:DACFORMAT	Set to match the selected DAC's SPI format	13.3
SYSDAC:NVMODE	Select the most appropriate behaviour on invalid measurements	13.8
SCW:GO	Set to 1 for each active sensor	10.10
SCW: CONT	Set to 1 for each active sensor to operate continuously	
SCW:INCE	Set to 0 to operate Type 2 or Type 6 sensors in Absolute Mode	11.9
SCW:SIE	Set to 0, unless a spare IO is used as a sample indicator, e.g. to activate after a DAC update	10.18
SIC		
LEDCW: LEDMAP	Set to IO1=1, IO5=5 or IO6=6 to drive an LED	13.11
LEDCW:LEDTHRESHOLD	Set to the desired Amplitude threshold if LED control is required	
DACPOSA…D	Defines the positions of the start and end of the up and down slopes	13.7
DACLIMX,Y	Set to the minimum and maximum DAC values required when VALID	
DACCW:DACEN	Set to 1 for each sensor that generates a DAC output	13.3, 13.10
DACCW:DACOP	Setting depends on the DAC SPI data format and timing of updates	
DACCW:DACADDR	Defines which output of a multi-output DAC each sensor updates	
DACCW:DACNV	Sets value of DAC output when a sensor's measurement is invalid	13.8
STYPE:TYPE	Set sensor 1's TYPE field to the type of sensor connected to the CTU	10.17, 11.2
STYPE:SUBTYPE	Set each sensor's SUBTYPE field to match the sensor's Subtype	
PTAH:HYSTERESIS	Set to 0 unless hysteresis is required in the DAC output	13.6
SAVEKEY SYSCW:SAVE	Once all other registers are programmed, set SAVEKEY to 0x0C1C and write a 1 to the SAVE bit to set configuration as the non-volatile default	11.11

13.13 Layout and Filtering Considerations

The noise present at the output of the DAC is a combination of the CTU's own measurement noise, noise that is introduced into the analog output of the DAC by the DAC itself, and system interference. This interference can be mimimised by following the DAC manufacturer's circuit and layout recommendations. Coupling mechanisms include the power supply. Care should be taken to isolate the CTU and DAC power supplies sufficiently, since the CTU includes significant digital processing and its excitation circuit handles AC current.

It is possible to further reduce the noise with appropriate filtering, for example using an RC filter illustrated in Figure 46. Rfilt and Cfilt are the filter resistor and capacitor. Rload represents the circuit's load, for example an amplifier. The filter capacitor's 0V connection is shown close to the load to minimise noise coupled along the 0V connection.

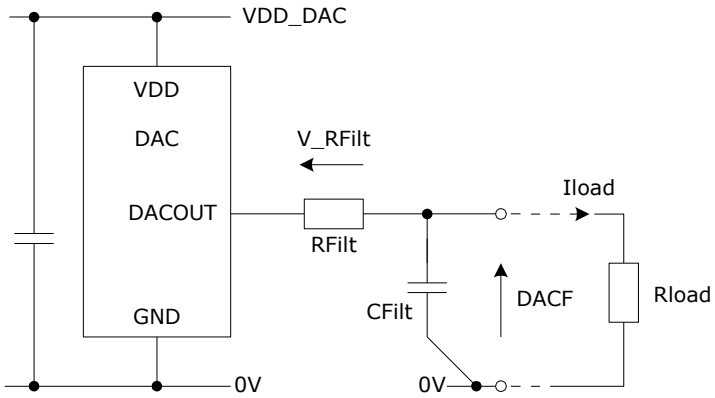


Figure 46 simple filter for DAC output

The filter bandwidth can be chosen to pass the signal of interest representing position. With a maximum update rate on the order of 2000 samples per second, a filter bandwidth of 1kHz will typically make little difference to the dynamic performance. The other consideration is loss of accuracy due to load current Iload flowing through filter resistor Rfilt. If the load resistance is 1Mohm or greater, Rfilt can be 1kohm to yield 1LSB error in 10 bits, and Cfilt 150nF to yield 1kHz filter bandwidth.

Active filters and signal conditioning circuits may be required for more demanding applications, for example when the product's output is distributed over an interface cable and is subject to EMI.

13.14 DAC Related Specifications

Table 44 lists specifications that relate to the CTU's DAC driving function.

Table 44 DAC related specifications

Item	Min	Typical	Max	Units
SPI clock frequency		3.2		Mbit/s

The rate at which each DAC output is updated depends on the type of sensor, the number of sensors connected and the SYSI setting (section 10.2). Table 45 lists sample rates with SYSI set to 0 to yield the highest possible rate.

Table 45 DAC update rate (SYSI=0, INCE=0)

Sensor Type	Number of active sensors	Minimum update rate	Units
4	1	2400	Samples/s
4	2	1200	
2	1	1500	
3	1	470	
6	1	1500	

14 Bootloader Operation

14.1 Overview

The CAM312 chip's internal software is partitioned into two fields: Application Code and Bootloader Code. The Application Code is responsible for normal CAM312 chip operation including measurements, with communication through the register interface described in section 11. The Bootloader Code can be used to update the Application Code using the CAM312 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 (section 10.8). The version number of the Bootloader Code can be read from the BOOTVER register (section 10.7).

The procedure for uploading new Application Code over the CTU chip's SPI interface is specified in this section. These details will be required if the host processor is required to perform the update process. The upload procedure may alternatively be performed by a PC communicating with the CAM312 chip over the SPI interface using CambridgeIC's CTU Adapter.

14.2 Applications

New versions of Application Code are typically released to enable new features. For example support for new Sensor Types or communication with external peripherals. Instead of buying a new stock of chips, a customer can program existing chips with the new code.

14.3 CTU Firmware File Format

New CAM312 Application Code is provided in the CTU Firmware File format (.cff). This comprises a Header Block, a Data Block and a Checksum Word. The Header Block is plain text (each byte representing a text character using ASCII codes), while the Data Block and Checksum Word are binary.

The Header Block can be viewed on a PC as a text file, for example using *WordPad* or dedicated hex and binary code editing software such as *Hex Editor Neo*. The text includes version number, build date and a description of the CRC algorithm that can be used to checksum the Data Block.

The Header Block comprises rows that start with the "#" character and end in line feed ("<LF>"). The data block can be identified by searching for the character after <LF> that is not "#". In byte representation, this means the bytes that follow the first occurrence of 0x0A that is not immediately followed by 0x23.

The Data Block comprises the remaining bytes in the .cff file. The last two bytes are the Checksum Word for the Data Block. This may be used to verify that the Data Block is valid and uncorrupted. If the Data Block has an odd number of bytes, pad with 0x00 at the end so that it includes only complete words.

14.4 Bootloader Process

The Bootloader works with the process illustrated in Figure 49. The first step is to identify whether or not the CAM312 chip already has valid Application Code, because this determines which Bootloader entry method to use.

If the CAM312 does include Application Code already, or unsure, the host must use the "BOOTLOAD Bit Entry Method". The host must first set the BOOTLOAD bit in the SYSCW register (section 10.1), using a normal SPI transaction as specified in section 9. Following that SPI transaction, the nSS line must remain high for at least a time TnRST2nSS illustrated in Figure 47 and specified in Table 46.

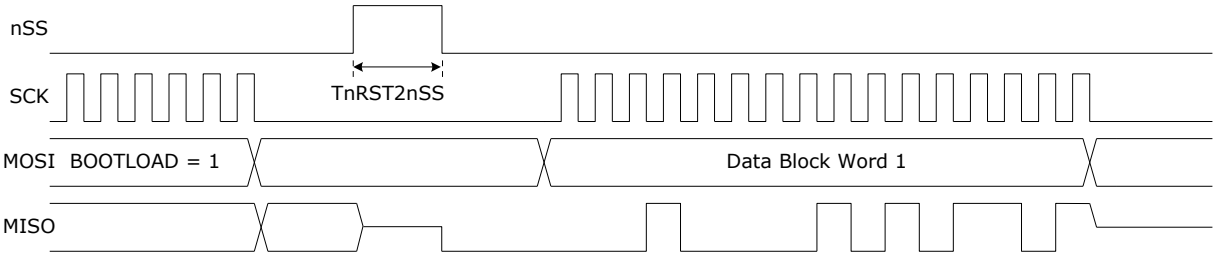


Figure 47 timing of first Data Block Word SPI transaction after setting BOOTLOAD=1

If the CAM312 does not include Application Code, the host must use the "Reset Entry Method". This begins with the host resetting the CAM312. This must be done either by pulling nRST low or by removing power (VCTU=0V) then reapplying it. The CAM312 chip's nSS line must be pulled high during or after reset, and must be pulled low before the

first SCK transition. The time between coming out of reset and the nSS low edge before the first SCK edge is denoted TnRST2nSS. The minimum value of TnRST2nSS is specified in Table 46.

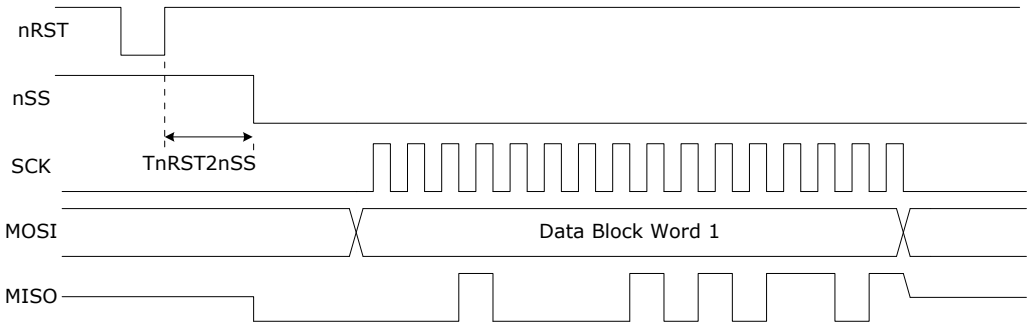


Figure 48 timing of first Data Block Word SPI transaction after reset

The host must now perform the Data Block Transfer Process, as detailed in section 14.5. There are 4 possible results: "successful", "CTU chip not in Bootloader Mode", "Unexpected CTU Response" or "Check CTU Wiring".

If successful, the host may send the word 0xB002 to the CAM312 over SPI to request a reset. It must then wait for internal validity checks to complete, like after any reset (section 11.1). Once complete, the CAM312 chip is ready for normal operation running the new Application Code.

If the CAM312 chip returns the value of the SYSID register as the first word of the Data Block Transfer Process (factory default 0xABCD) then it is still running old Application Code, and is not in Bootloader Mode. The upload may either be aborted or begun again.

If the Data Block Transfer Process fails with a MISO error (MISO equal to 0x0000 or 0xFFFF) then the CTU chip's wiring should be checked. In particular its power supply, reset circuitry and SPI connections.

If the CAM312 chip returns an unexpected response over SPI during the Data Block Transfer Process (typically 0x0BAD) then the Bootloader process has failed and must be restarted. Sending 0xB002 to the CAM312 over SPI may initiate a reset. Once validity checks have been completed (section 11.1) the CAM312 will return the value of the SYSID register if the host sends 0x0000 over SPI, but only if the CAM312 chip still includes old Application Code. If not, the CAM312 chip does not include valid Application Code any more, and the Bootloader Process must be repeated.

When implementing the Bootloader process, it is essential to implement both the BOOTLOAD Bit Entry Method and the Reset Entry Method. The Reset Entry Method can not be used when starting with valid Application Code, so the BOOTLOAD Bit Entry Method must always be implemented. However there is always a risk of an unexpected behavior of the host system, for example another device sharing the same SPI device unexpectedly accessing the SPI bus during Bootloader data transfer. This may corrupt CAM312 Application Code, leaving the part with no valid Application Code. The Reset Entry Method always allows new Application Code to be programmed in this situation.

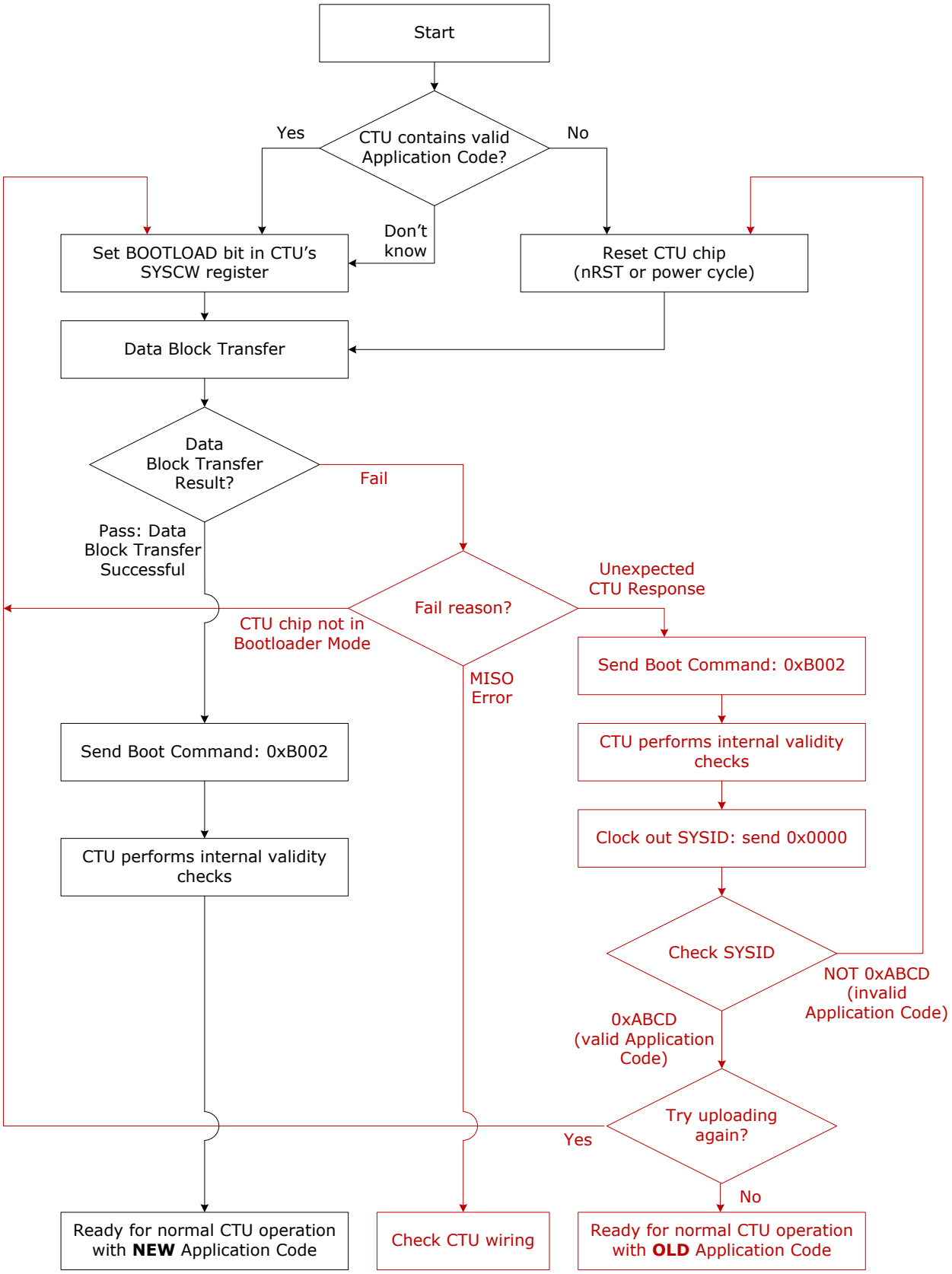


Figure 49 Bootloader flowchart

14.5 Data Block Transfer Process

The Data Block Transfer process is for sending new Application Code to the CTU, and is illustrated in Figure 50. It forms part of the Bootloader Process described in section 14.4.

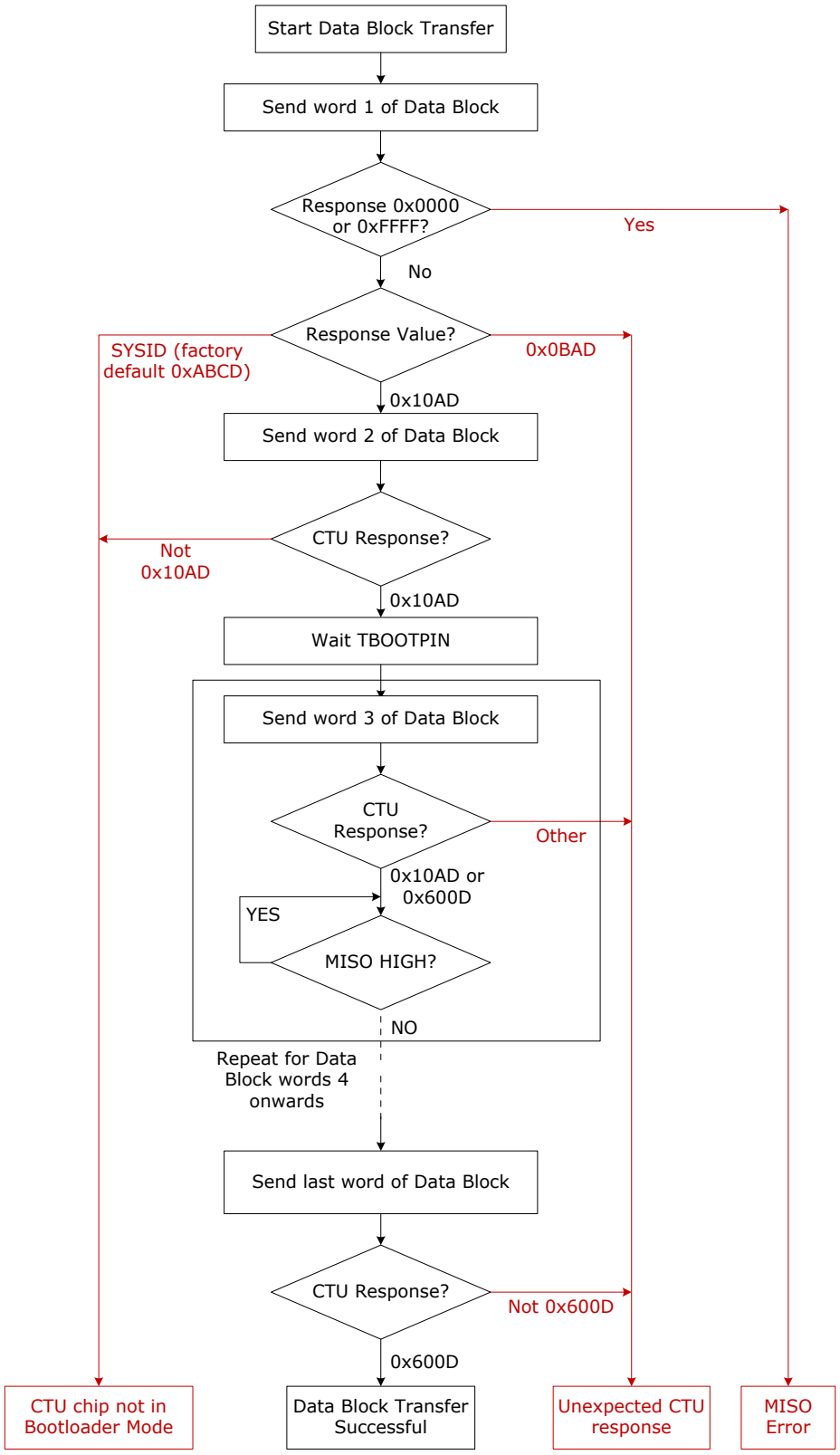


Figure 50 Data Block Transfer Process

Data is transferred in 16-bit words over the SPI interface, as described in section 14.6. Each time the host sends a word, the host should check the response received on the CTU chip's MISO output…

- If the response is 0x0000 or 0xFFFF then there is a MISO Error. Possible causes include a wiring fault, the CTU chip being unpowered, or the CTU chip being unexpectedly held in reset.
- The responses 0x10AD indicates the data transfer process is proceeding successfully.
- The response 0x600D indicates the data transfer process has completed OK.
- If the response is equal to the CTU chip's SYSID value (0xABCD by default) then the CTU chip is not in bootloader mode. Data Block Transfer was unsuccessful and should be terminated. The CTU chip contains valid Application Code. This means that the BOOTLOAD Bit Entry Method should have been successful, and can be retried.
- If the response is 0x0BAD or another response not listed above then the Data Block Transfer process has failed unexpectedly. Possible causes include Data Block data not valid for this CTU chip, the host sending a data word over SPI too early, SPI interface interference and CTU power supply voltage out of specification.

The host must send the first two words of the Data Block and then pause before the third. The length of the pause is measured from the last SCK edge of the second word and the first SCK edge of the third word, and is denoted TBOOTPIN. A minimum value of TBOOTPIN is required, and it is specified in Table 46. This pause is for Bootloader initialisation.

The remainder of the Data Block is then sent to the CAM312 chip, one word at a time. The host must check the state of the MISO line after each word. If it is high, the host must pause until MISO has gone low. The time taken for MISO to go low is denoted TBOOTWAIT. Its maximum value is specified in Table 46.

The CAM312 chip's response to the last word of the Data Block will be 0x600D if successful.

14.6 SPI Communication with the CTU in Bootloader Mode

Communication with the CTU in Bootloader Mode uses the same SPI mode and bit ordering as described in section 9. The minimum clock low and high times and the clock period (TSCKL, TSCKH, TSCK) are also the same.

The CAM312 chip's SPI select line nSS must be low while data is being clocked into the CAM312 chip. nSS may remain low for the entire Bootloader process. Alternatively, nSS may be pulled high after any Data Block Word, providing it is pulled low again before the next.

As noted in section 14.5, the host must check the state of the MISO line after sending each Data Block Word. If MISO is high, the host must pause communications until the CAM312 pulls it low. The time between the last SCK low going edge of a Data Block Word and MISO going low is denoted TBOOTWAIT. The maximum value is specified in Table 46.

Note that the CAM312 chip only pulls MISO low to indicate it is ready for the next Data Block Word if the nSS line is low. If nSS is high, MISO will be high impedance, to allow other slave devices to share the same SPI bus. If the host chooses to pull nSS high after a Data Block Word, it may test the state of MISO by pulling nSS low again. Providing SCK remains low throughout, this test may be repeated until the host detects MISO is low.

A time delay TMISOL2SCKH_BOOT is required each bootloader SPI transaction, between the time MISO goes low after one SPI transaction and when the host drives SCK high for the next SPI transaction. This is illustrated in Figure 51. The minimum value of TMISOL2SCKH_BOOT is specified in Table 46.

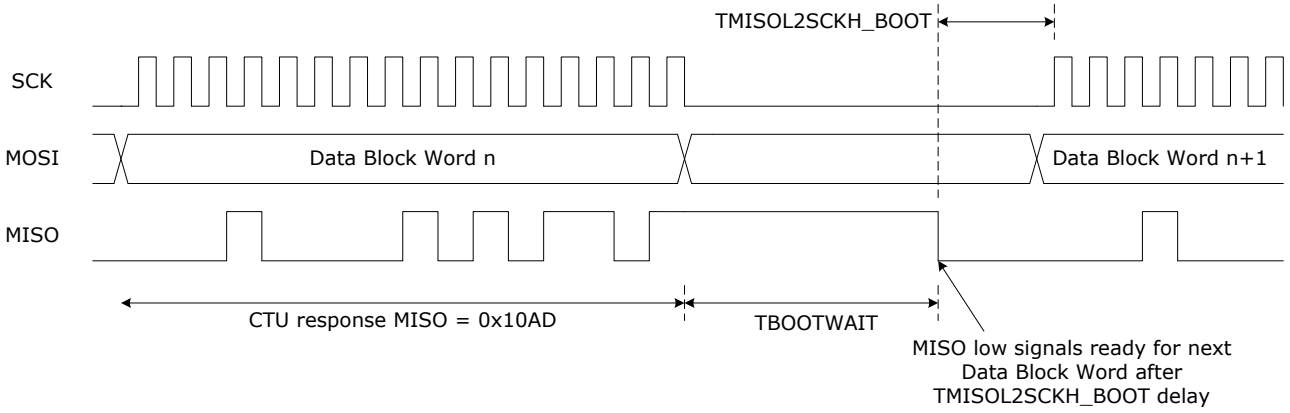


Figure 51 SPI communication, sending Data Block

14.7 Bootloader Timing Specifications

The following parameters are the same for Bootloader SPI as for normal operation, and are specified separately in Table 20: TSCKL, TSCKH, TSCK, TSCKR, TSCKF, TMISOR, TMISOF, TMOSIV2SCKH, TSCKH2MOSIX, TnSSL2MISOV, TSCKL2MISOV, TnSSL2SCKH, TSCKL2nSSH, TnSSH2MISOZ.

The specifications for the Bootloader timings referred to above are in Table 46.

Table 46 Bootloader timings

Parameter	Description	Min	Max
TnRST2nSS	Delay between end of reset and first SCK high	3.2ms	
TBOOTPIN	Pause required between 2nd and 3rd Data Block words	500µs	
TMISOL2SCKH_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	

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 46 specifies a minimum value, based on a host responding to MISO low immediately and operating with the minimum TMISOL2SCKH_BOOT value. TBOOTLOAD(min) is the time the bootload process takes for a sufficiently fast and well optimised host.

15 Package Details

15.1 CAM312ME

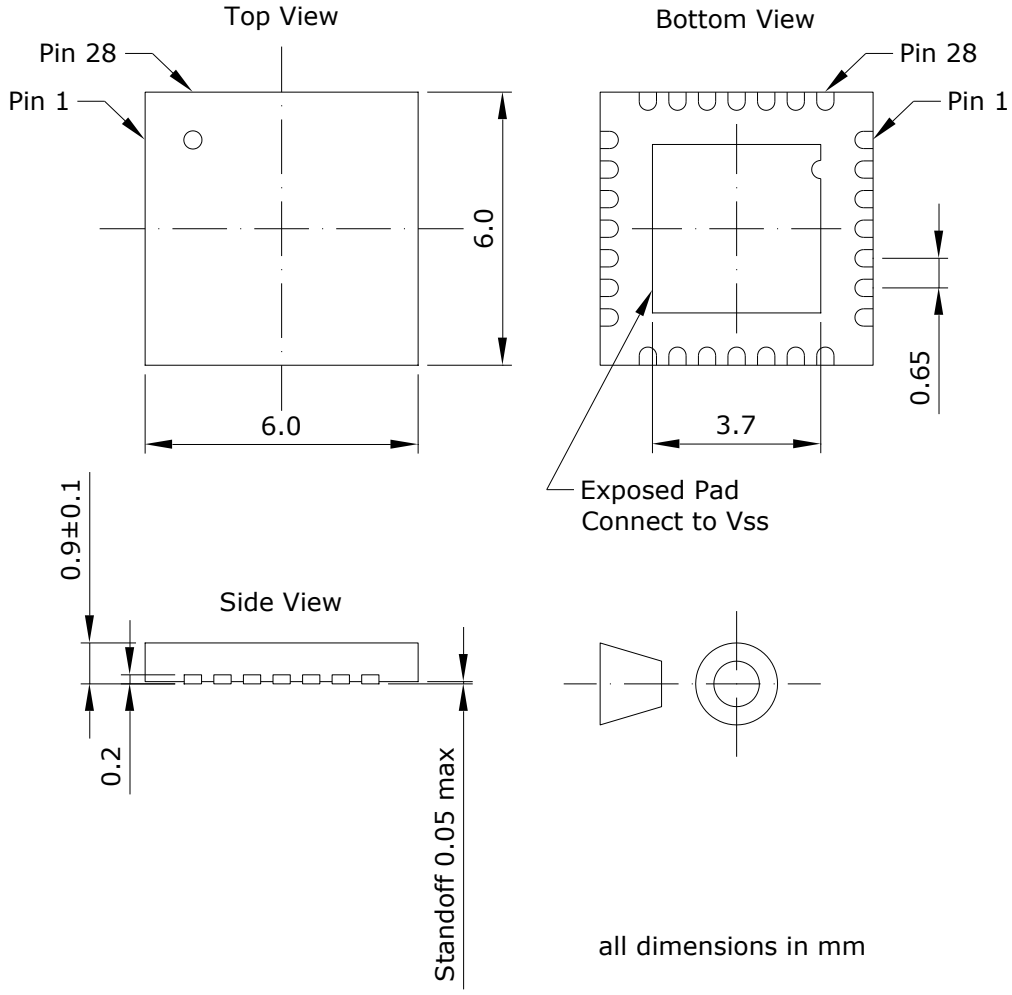


Figure 52 CAM312ME plastic quad-flat no-lead 28-pin (QFN: M suffix)

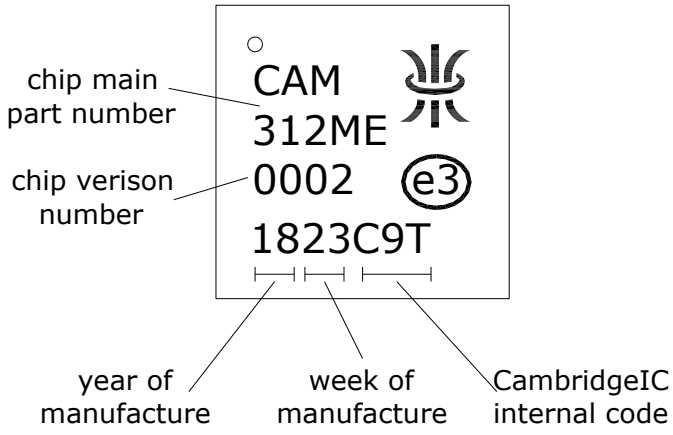


Figure 53 CAM312ME product markings

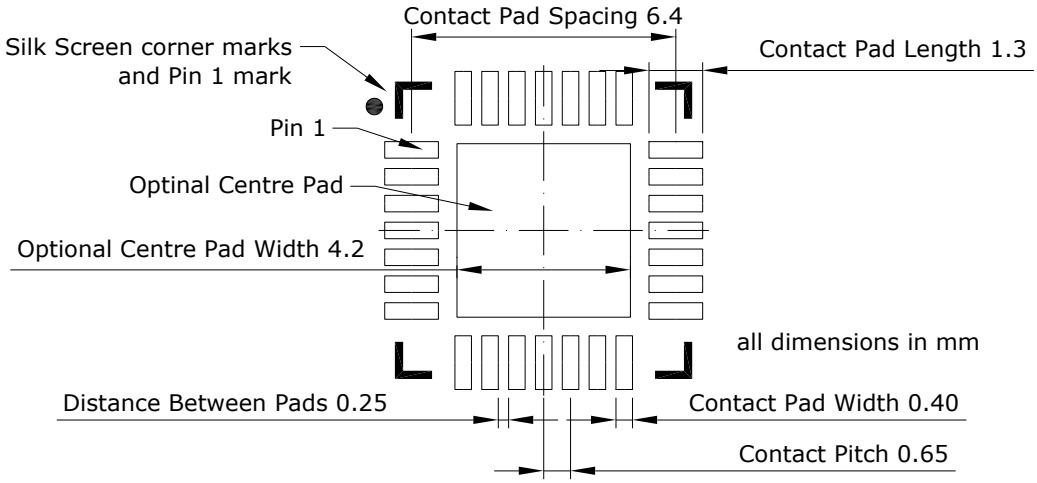


Figure 54 CAM312ME Recommended PCB Footprint

16 Tape and Reel Specifications

16.1 CAM312ME

CAM312ME chips are available in tape and reel on complete reels of 1600 parts. The carrier tape is illustrated in Figure 55, and dimensions are specified in Table 47.

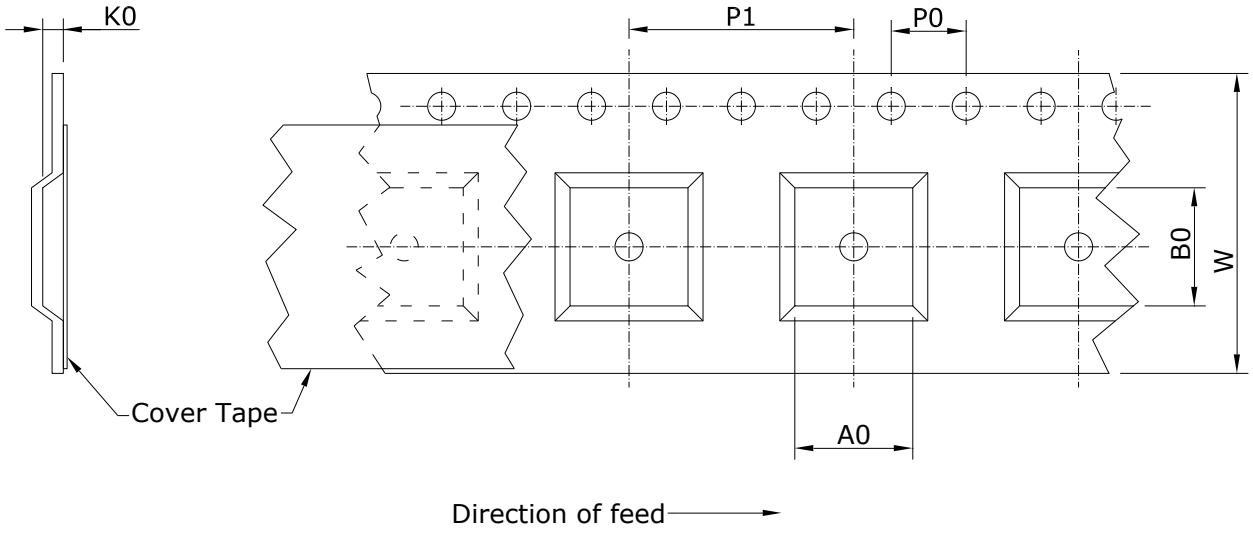


Figure 55 CAM312ME carrier tape dimensions

Table 47

Tape and Reel Specifications		Dimensions in mm						
Package	**Units per reel**	**Reel diameter**	**W**	**P0**	**P1**	**A0**	**B0**	**K0**
28-pin QFN	1600	330	16	4	12	6.3	6.3	1.1

17 Reflow Soldering Recommendations

The CAM312 is available in lead free packaging only. The recommended reflow soldering temperature profile is illustrated in Figure 56. Values are shown in Table 48.

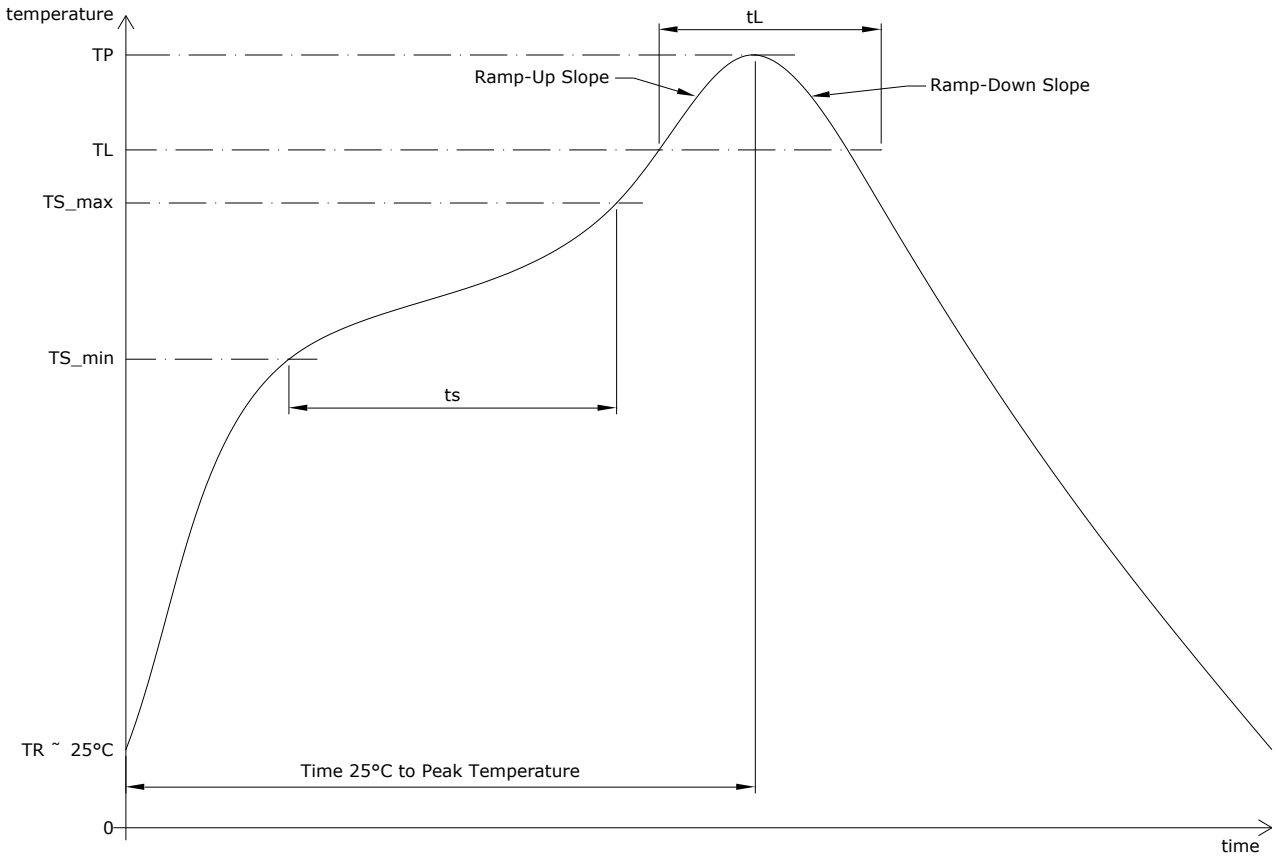


Figure 56 reflow soldering temperature profile definitions

Table 48

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		

18 Environmental

Table 49

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

19 RoHS Compliance

The CAM312 uses Matte Tim (Sn) pin finish. CambridgeIC certifies, to the best of its knowledge and understanding, that the CAM312 chip is in compliance with EU RoHS directive 2011/65/EU and 2015/863.

20 Document History

Table 50 main changes

Rev	Date	Comments
0006	17 March 2020	Increased measurement times based on updates in Application Code V1.04 Added DAC update rates.
0007	17 January 2022	Corrected CAM312 part number on this page Added TnRSTL parameter for reset pin low duration Clarified where registers and bit fields are not implemented Corrections to SPI timing parameter names Type 2 schematic not suitable for repeated low power mode measurements, added notes to contact CambridgeIC for alternatives Updates to power down process including new timing parameter TnSSLW2nSSLP and requirement to repeat measurements until VALID after recovery from power down
0008	16 May 2022	Corrected C_FCOS1, C_FSIN1 component designations in Figure 12 Removed TWW_BOOT bootloader timing parameter, replaced with new timing parameter TMISOL2SCKH_BOOT specifying time delay from MISO low to next nSS high. Updated TBOOTLOAD(min) to 1.4s
0009	22 Sept 2022	Updated bootloader flowcharts and text to include exit on MISO error
0010	25 Sept 2023	Noted that the CAM312 drives MISO high when in low power mode. Corrected C_B location in figure "Sensor coil and filter connections to CAM312"
0011	26 Sept 2024	FDY4000CZ now end of life, replaced with Si1016CX. R_DRIVEL changes to 1kΩ as well.

21 Contact Information

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

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

22 Legal