Description

CambridgeIC's CAM502 Central Tracking Unit (CTU) IC is a single-chip processor for precise position measurement. It implements the electronic processing for resonant inductive position sensing technology. The CAM502 is optimised for high-speed applications.

The CTU measures the position of a contactless, inductively coupled target relative to a sensor that is built from a printed circuit board to CambridgeIC's design.

Features

- Resonant inductive position sensing engine
- Fully ratiometric measurements
- Automatic tuning to target frequency
- SPI communications (slave device)
- Pipeline measurement for high-speed operation
- Control input for synchronising measurements
- User IOs for position triggers and sample indicators
- Internal software upgradable over SPI
- Works with Type 2 and 6 sensors
- External crystal controlled timing

Performance

- Group Delay physical position to SPI ready ≤140µs
- Up to 5100 independent samples per second
- SPI interface bit rate up to 10Mbit/s
- Up to 15 bits noise free resolution without filtering
- Up to 18 bits noise free resolution with filtering

Applications

- Position feedback for high-speed motion control

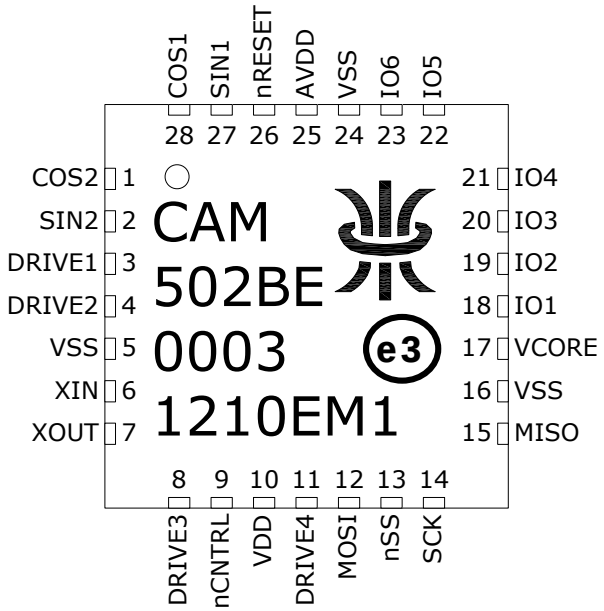


Figure 1 CAM502 28-pin QFN pinout

Product identification	
Part no.	**Description**
CAM502BE-0003	CAM502BE in 28-pin QFN -40°C to +125°C

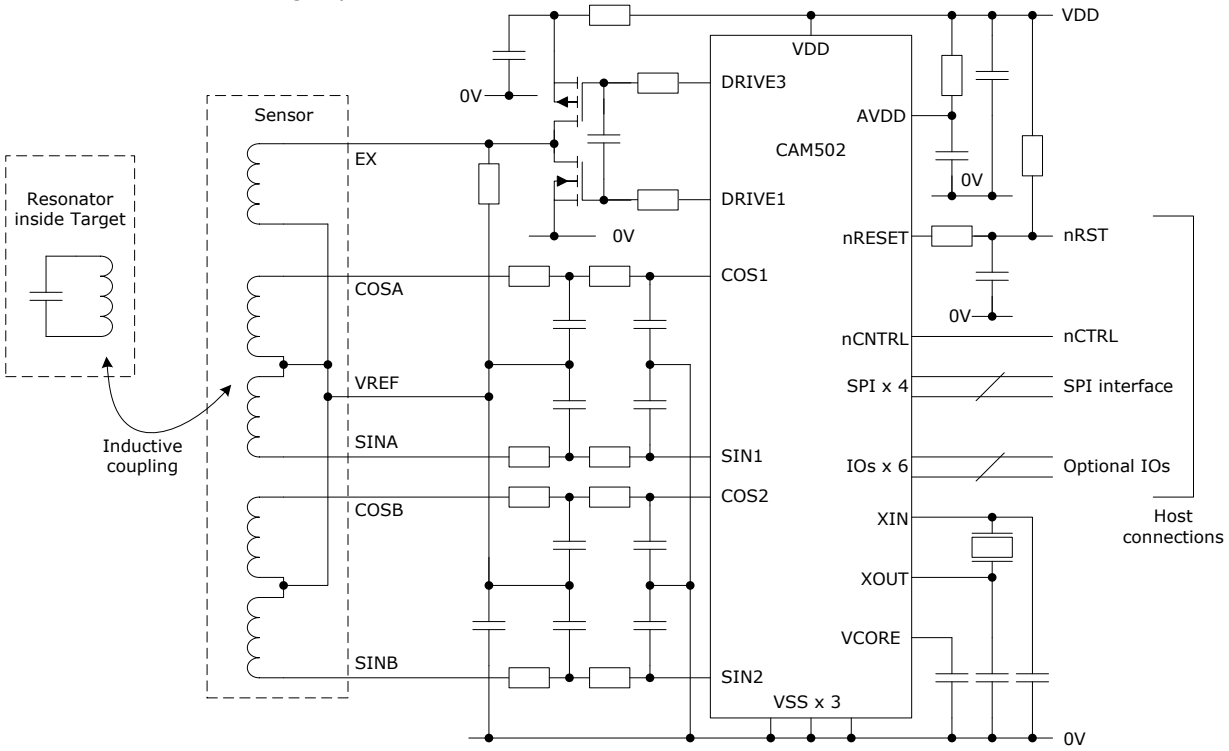


Figure 2 Circuit for reading a Type 6 sensor

1 CAM502 Functional Description

1.1 Overview

The CAM502 Central Tracking Unit (CTU) works with a sensor built from a PCB to measure the position of a contactless target. Targets comprise an inductively coupled resonant circuit. Sensors are available for linear and rotary measurement and in a range of different sizes.

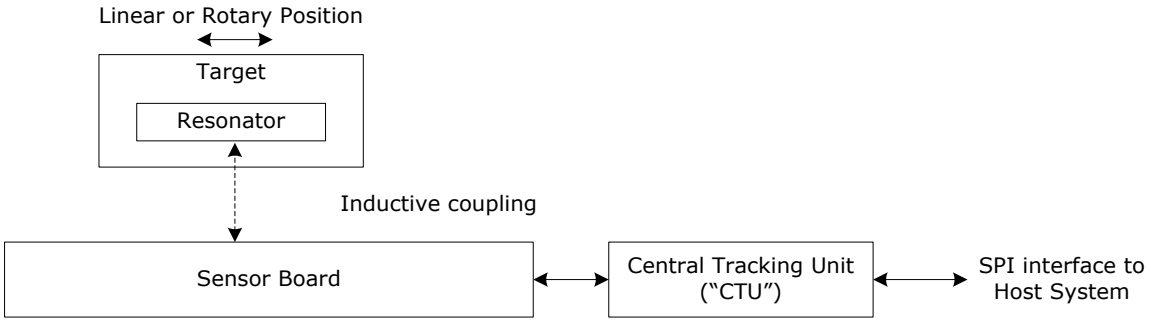


Figure 3 resonant inductive position sensing system

Sensor Types differ in their details, but the same basic measuring principle applies to them all. The CTU 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 CTU 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.

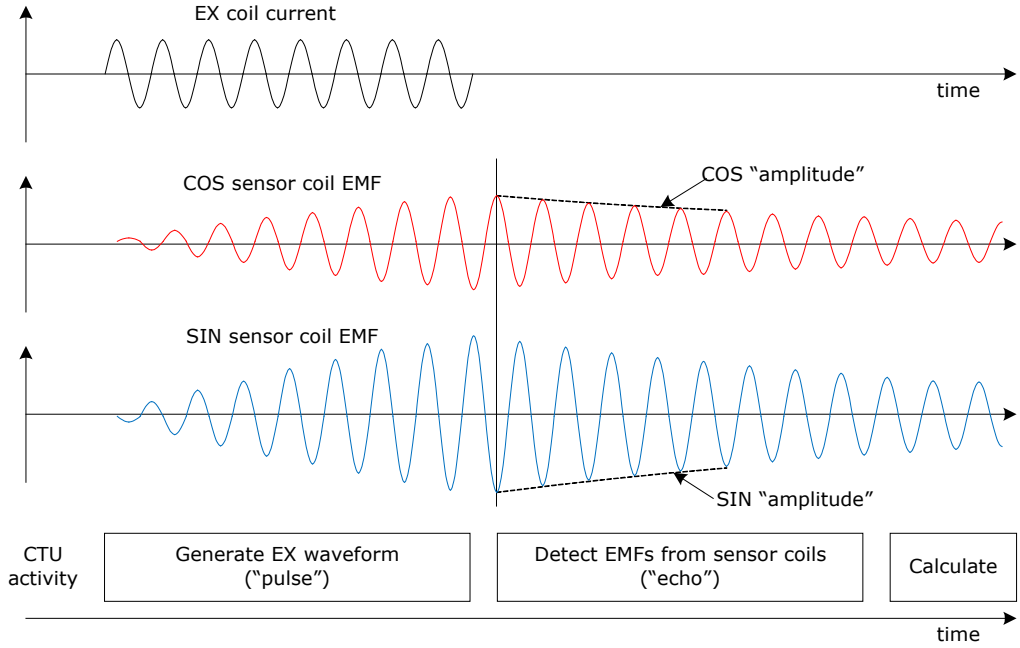


Figure 4 Electronic measuring process

1.2 Sensor Types

Table 1 lists the sensor types currently supported by the CAM502, and includes section references for more details on each Type.

Table 1 Comparison of sensor types

Type	Number of coils in each sensor		Application	Max number of sensors (CAM502)	Has Subtypes?	Section number for details
	Excitation	Sensing				
2	1	4	Long linear sensors	1	No	3
6	1	4	Precision rotary sensors	1	Yes	4

1.3 Pin Names and Functions

Table 2 summarises CAM502 pin functions and type. Please refer to section 2 for electrical characteristics for each type.

Table 2

Signal Name	Type	Function
VDD	Power	Positive supply voltage.
AVDD	Analog Input	Analog supply voltage, decoupled from VDD.
VSS (x3)	Power	0V connection and common return for sensor inputs.
VCORE	Power	Output of on-chip 2.5V regulator, requires external decoupling capacitor to VSS.
nRESET	Digital Input, 5V Tolerant	Hardware reset, active low
nSS	Digital Input, 5V Tolerant	SPI Interface line: Slave Select, active low.
SCLK	Digital Input, 5V Tolerant	SPI Interface line: Serial Clock.
MOSI	Digital Input, 5V Tolerant	SPI Interface line: Master Out, Slave In.
MISO	Digital Output, 5 V Tolerant (1)	SPI Interface line: Master In, Slave Out.
IO1 and IO2	Digital or Open Drain Output, 5V Tolerant	User configurable IO.
IO3, IO4, IO5 and IO6	Digital or Open Drain Output	
nCNTRL	Digital Input	Control pin, optionally used to synchronise measurements
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.
XIN, XOUT	Analog	Connections to external crystal
Exposed Pad	Shield	Large pad under package (Figure 45). Connect to Vss.

Note (1): MISO is driven as a digital output by the CAM502 during SPI transactions (nSS low), and is Open Drain at other times 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, CAM502BE
VDD start voltage relative to VSS		0V	For reliable power on reset
VDD rise rate relative to VSS	0.1V/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	-0.3V to +5.6V
Voltage on any other pin relative to VSS	-0.3V to (VDD+0.3V)
Current into or out of Digital Output	4mA

2.3 Digital Input Specifications

Table 5 digital input specifications

Item	Min	Max
Input Low	VSS	0.2 x VDD
Input High, 5V Tolerant	0.7 x VDD	5.5V
Input High, NOT 5V Tolerant	0.7 x VDD	VDD
Input leakage current		±4µA

2.4 Digital Output Specifications

Table 6 digital output specifications

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

2.5 Application Memory Characteristics

The CAM502 includes a processor with FLASH memory. This can be updated with new Application Code over its SPI interface, see section 10. It can also be updated with new Configurable Defaults, see section 9.13. FLASH memory related specifications are in Table 7 below.

Table 7 application memory characteristics

Item	Min	Max	Comments
Number of FLASH updates		2000	Across Operating Supply Voltage and Operating Temperature
Retention time	20 years		

3 Type 2 Sensor Application

Type 2 sensors are for the precise measurement of linear position. Please refer to sensor datasheets for more information.

3.1 Circuit Schematics, Type 2

Figure 5 shows how the CAM502's supplies should be connected, the chip's connections to the host and to the excitation and sensing circuits. Component values are listed in Table 8.

VCORE is an external connection to an on-chip 2.5V regulator. This pin requires an external decoupling capacitor. It must not be connected to other circuitry.

AVDD is the analog supply input to the CAM502, and is used as a reference voltage. The values of R_AVDD and C_AVDD are important for the CAM502 to operate to full specification.

nRESET 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 (section 10).

MISO requires a pull-up resistor of 4.7kΩ if this is not provided by the host.

The CAM502 uses an external crystal, X1, to derive precise timing.

The nCNTRL pin is for measurement control. It must be connected to nSS in pipeline mode with CONT=0 (section 9.7) so that measurements are synchronised to the host's acquisition of position data over the SPI interface. nCNTRL is not 5V Tolerant, unlike nSS and the other SPI inputs to the CAM502. If the host outputs 5V logic signals, the level translation circuit of Figure 9 is required.

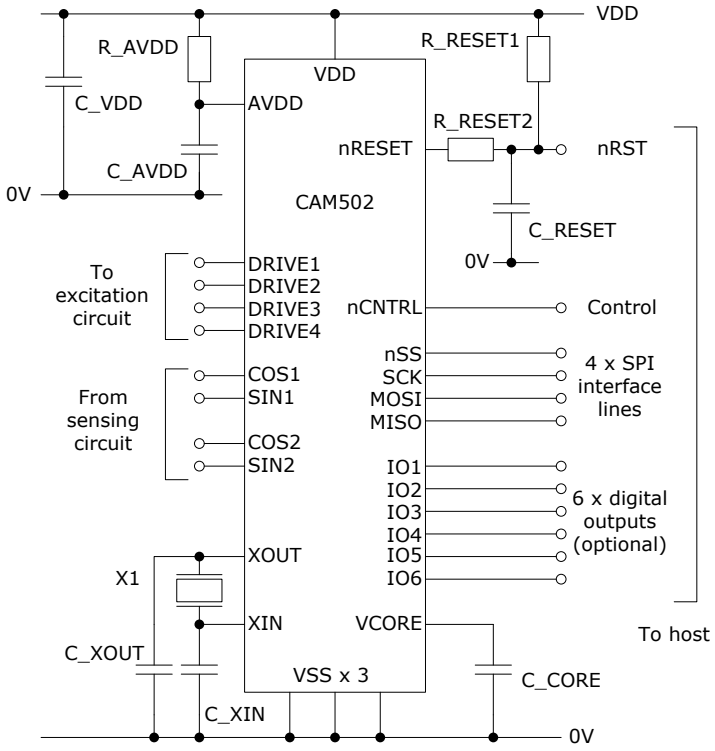


Figure 5 CAM502 connections, Type 2

The CAM502 uses an external MOSFET "H" bridge to drive a Type 2 sensor's excitation coil. Figure 6 shows the excitation circuit. MOSFET pairs Q1 and Q2 are each available as a single miniature device, see Table 8. The gate drive circuit uses 2 resistors and 1 capacitor per side, 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 capacitors C_Q1 and C_Q2 prevent excessive shoot-through current during switching. The network R_SN1, R_SN2 and C_SN absorb 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.

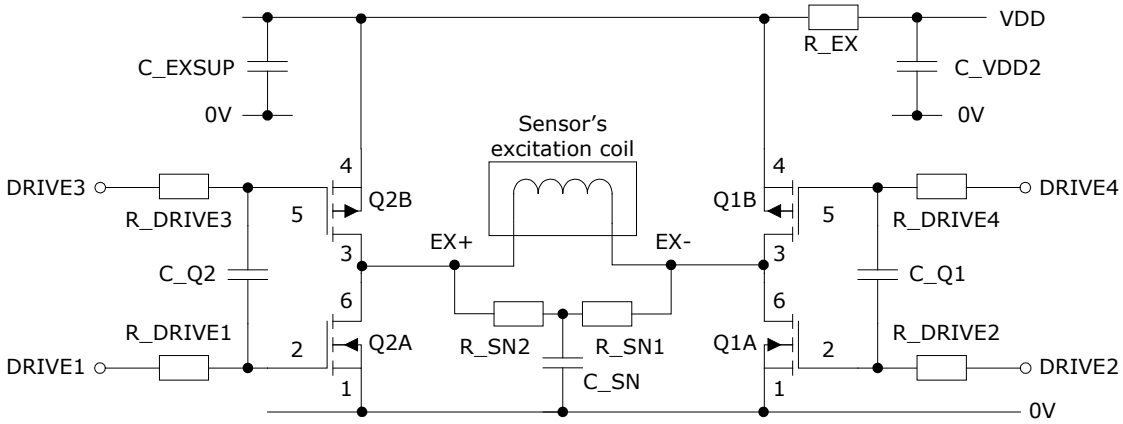


Figure 6 Excitation Circuit, Type 2

Figure 7 shows the components required for the sensor coil inputs to the CTU. Each of the 4 coil inputs has two stages of RC filtering formed by the components with prefix R_F2, C_F2, R_F and C_F. The reference voltage VREF is generated by the network R_BH, R_BL and CB. VREF should not be generated by other means.

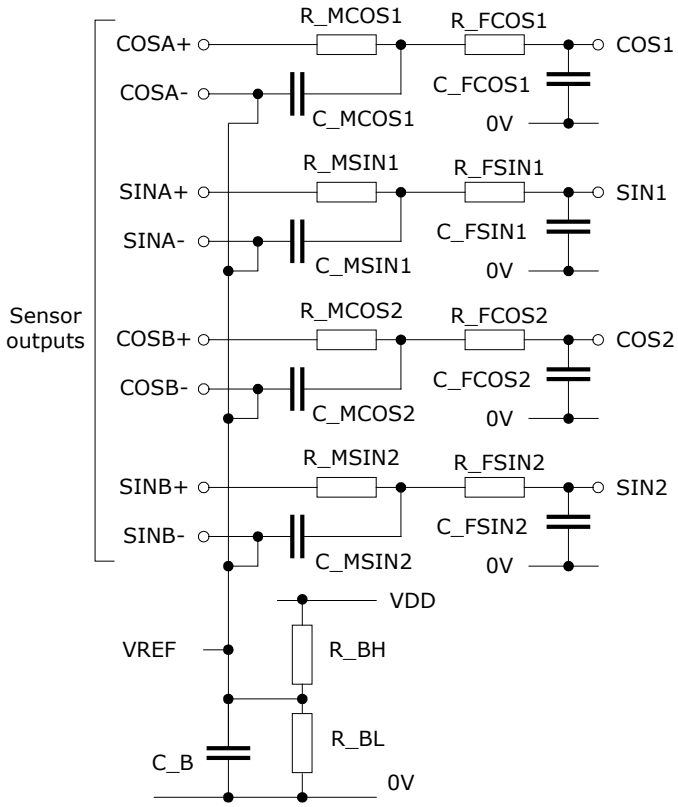


Figure 7 Sensor coil connections, Type 2

3.2 Components Required, Type 2

Table 8 lists the component values and numbers required for the schematics of section 3.1. The filter components connected to the CTU's sensor 1 inputs are important for reproducibility, and the table 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 more cost effective. Grade A and B reproducibility error is compared in Table 9.

Table 8 components required for a Type 2 sensor

Circuit Ref	Value	Tolerance		Number required
		Grade A	Grade B	
R_RESET1	10kΩ	±5%		1
R_RESET2	470 Ω	±5%		1
R_AVDD	10Ω	±5%		1
R_DRIVE1/2	1.6kΩ	±5%		2
R_DRIVE3/4	330Ω	±5%		2
R_EX	1Ω	±5%		1
R_SN1/2	100Ω	±5%		2
R_MCOS1/SIN1	100Ω	±0.1%	±1%	2
R_MCOS2/SIN2	100Ω	±1%		2
R_FCOS1/SIN1	1kΩ	±0.1%	±1%	2
R_FCOS2/SIN2	1kΩ	±1%		2
R_BH	15kΩ	±1%		1
R_BL	10kΩ	±1%		1
C_VDD	470nF	±10%		1
C_AVDD	10µF, ESR < 3Ω	±20%		1
C_CORE	10µF, ESR < 3Ω	±20%		1
C_EXSUP	100µF, ESR < 100mΩ	-50% / +100%		1
C_VDD2	100µF, ESR < 100mΩ	-50% / +100%		1
C_RESET	100nF	±10%		1
C_Q1/2	1nF	±10%		2
C_SN	10nF	±10%		1
C_MCOS/SIN1	2.2nF	±1%	±5%	2
C_MCOS/SIN2	2.2nF	±5%		2
C_FCOS/SIN1	220pF	±1%	±5%	2
C_FCOS/SIN2	220pF	±5%		2
C_B	470nF	±10%		1
C_XIN, C_XOUT	22pF (1)	±10%		2
X1	8MHz	±30ppm		1
Q1/2	FDY4000CZ			2

Note (1): For crystal X1 requiring 12pF load capacitance

Table 9 reproducibility error due to filter components

Grade	Reproducibility as % of Fine Pitch	Reproducibility in µm	
		Fine Pitch = 50mm	Fine Pitch = 100mm
A	±0.03%	±15µm	±30µm
B	±0.17%	±85µm	±170µm

4 Type 6 Sensor Application

Type 6 sensors are for the precise measurement of rotary position, and typically have a central opening for through-shaft operation. Please refer to sensor datasheets for more information.

4.1 Circuit Schematics, Type 6

The circuit for connecting a Type 6 sensor to the CAM502 chip is shown in Figure 8 below. Component values are listed in Table 10.

VCORE is an external connection to an on-chip 2.5V regulator. This pin requires an external decoupling capacitor. It must not be connected to other circuitry.

AVDD is the analog supply input to the CAM502, and is used as a reference voltage. The values of R_AVDD and C_AVDD are important for the CAM502 to operate to full specification.

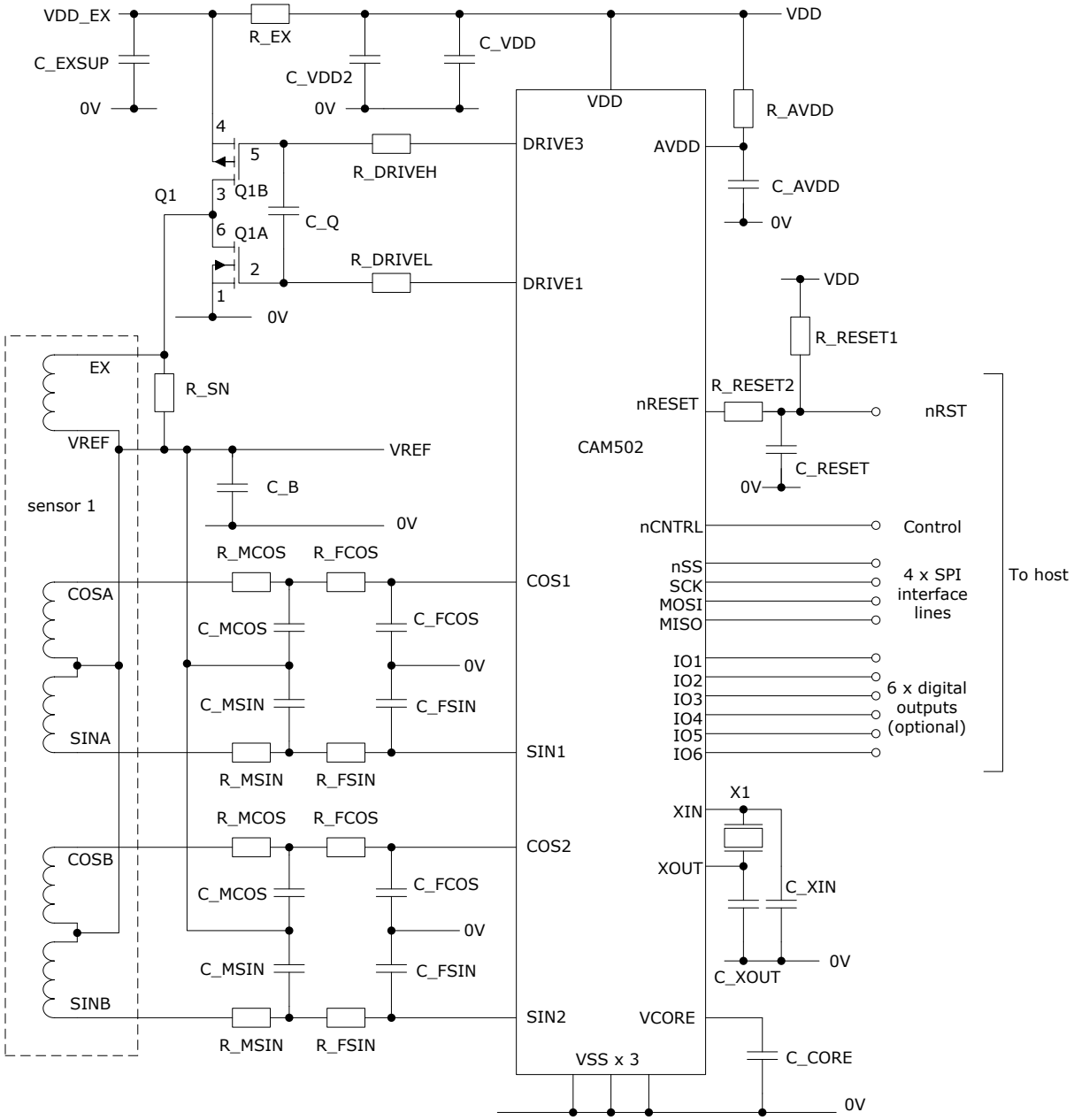


Figure 8 CAM502 connections, Type 6

The CAM502'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.

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

The CAM502 uses an external crystal, X1, to derive precise timing.

The nCNTRL pin is for measurement control. It must be connected to nSS in pipeline mode with CONT=0 (section 7.3) so that measurements are synchronised to the host's acquisition of position data over the SPI interface. nCNTRL is not 5V Tolerant, unlike nSS and the other SPI inputs to the CAM502. If the host outputs 5V logic signals, the level translation circuit of Figure 9 is required.

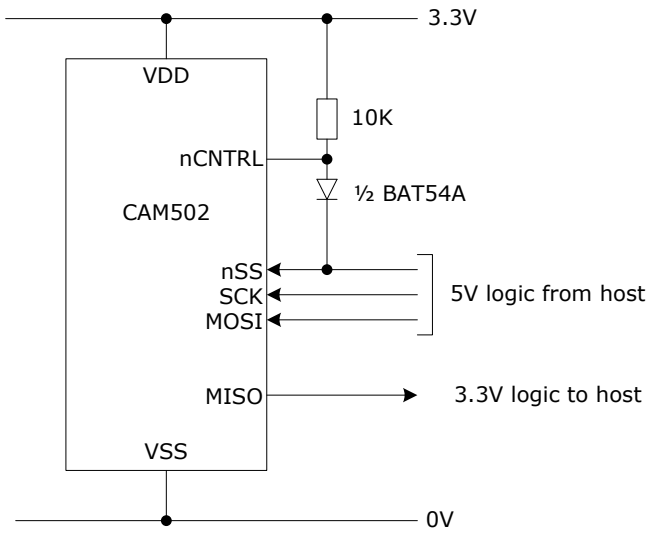


Figure 9 nCTRL level translation

The CAM502 uses an external MOSFET half bridge to drive a Type 6 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.

Each sensor's 4 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.

4.2 Components Required, Type 6

Table 10 lists the component values and numbers required for the schematics of Figure 8.

Table 10 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_AVDD	10Ω	±5%		1
R_DRIVEL	1.6kΩ	±5%		1
R_DRIVEH	330Ω	±5%		1
R_EX	1Ω	±5%		1
R_SN	1kΩ	±5%		1
R_MCOS1/SIN1	220Ω	±0.1%	±1%	2
R_MCOS2/SIN2	220Ω	±1%		2
R_FCOS1/SIN1	1kΩ	±0.1%	±1%	2
R_FCOS2/SIN2	1kΩ	±1%		2
C_VDD	470nF	±10%		1
C_AVDD, C_CORE, C_EXSUP, C_VDD2	10µF, ESR < 3Ω	±20%		4
C_RESET	100nF	±10%		1
C_Q	1nF	±10%		1
C_MCOS/SIN1	1nF	±1%	±5%	2
C_MCOS/SIN2	1nF	±5%		2
C_FCOS/SIN1	220pF	±1%	±5%	2
C_FCOS/SIN2	220pF	±5%		2
C_B	4.7µF	±10%		1
C_XIN, C_XOUT	22pF (1)	±10%		2
X1	8MHz	±30ppm		1
Q1	FDY4000CZ			1

Note (1): For crystal X1 requiring 12pF load capacitance

The filter components connected to the CTU's sensor 1 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 11.

Table 11 reproducibility error due to filter components, Type 6

Grade	Reproducibility due to filter components			
	As % of Fine Pitch (Sin Length)	Type 6.3 (Sin Length = 120°)	Type 6.5 (Sin Length = 72°)	Type 6.6 (Sin Length = 60°)
A	±0.03%	±0.036°	±0.022°	±0.018°
B	±0.17%	±0.20°	±0.12°	±0.10°

4.3 Supply Current

The CAM502's supply current depends on the rate at which it is sampling position, and the type of measurement. Pipeline measurement, described in section 9.7, yields the highest sample rates and the highest current consumption. Repeated single shot measurement (section 9.6) is suited to lower sample rates and yields lower current consumption. Single shot current consumption also depends on whether measurements are absolute (INCF=0) or incremental (INCF=1), see section 9.8.

Figure 10 is a graph of supply current against sample rate for the different modes, operating at a supply voltage of 3.3V. Figures include both the CAM502 supply current and the current required for the excitation circuit.

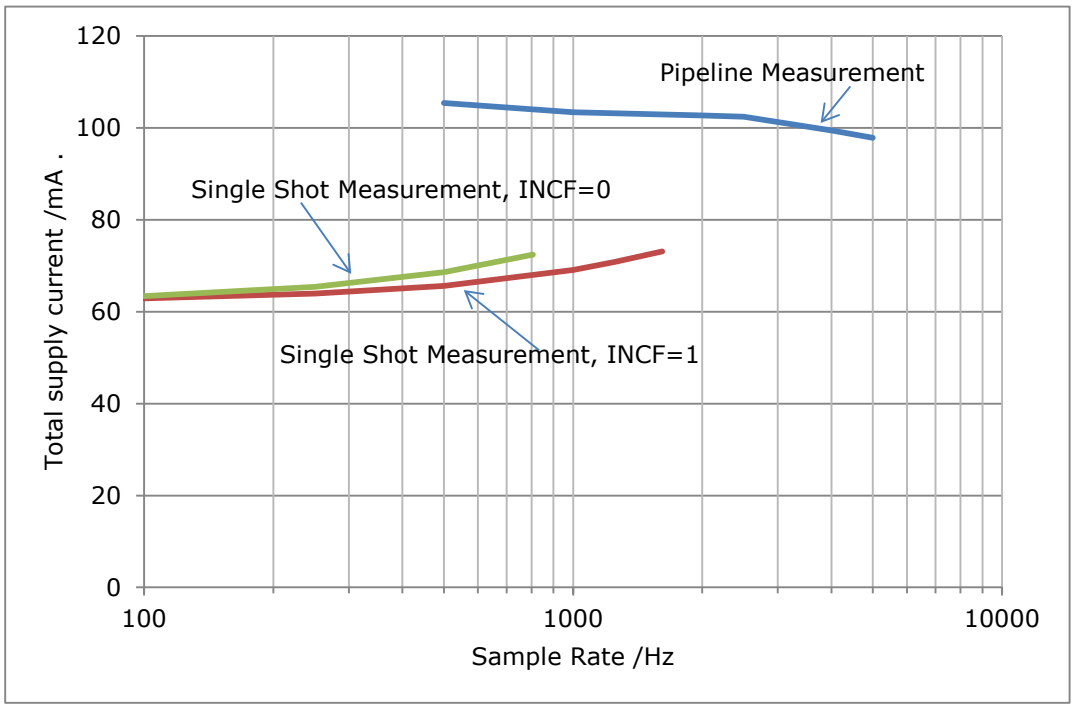


Figure 10 Supply current as a function of sample rate, at 3.3V

4.4 Noise Free Resolution

Position resolution is not limited by the number of bits reported over the SPI interface (≥18 bits for a Type 6 sensor). A more useful measure of resolution is Noise Free Resolution. This describes how many different positions the CAM204 can distinguish, considering the random noise present in the lower bits of its output. It is generally a more useful measure than interface resolution for a system having excess interface resolution.

The random noise present in the CTU's reported measurements can be considered Gaussian (well behaved noise). There are two general measures of Random Noise, Peak to Peak Noise and Standard Deviation. Defining Peak to Peak Noise such that it encompasses 99.9% of samples (100% is physically impossible due to the statistical nature of noise) yields the following relationship:

$$Peak\ To\ Peak\ Position\ Noise = 6.6 \times Standard\ Deviation$$

Equation 1

Another common measure of noise used in encoders is Noise Free Resolution, which is related to Peak to Peak Noise as follows:

$$Noise\ Free\ Resolution = log_2\left(\frac{360°}{Peak\ to\ Peak\ Position\ Noise}\right)$$

Equation 2

Noise Free Resolution depends on the Amplitude reported by the CAM502, which is a function of gap from sensor to target and their metal environment. Reported Amplitude can be established from the relevant sensor datasheet, or by experiment. Noise Free Resolution is also a function of sensor Subtype, with sensors having a higher Subtype number (more fine track sinusoidal periods) delivering the better performance.

Figure 11 shows the expected Noise Free Resolution as a function of Reported Amplitude for each Subtype, for Single Shot measurement (section 9.6).

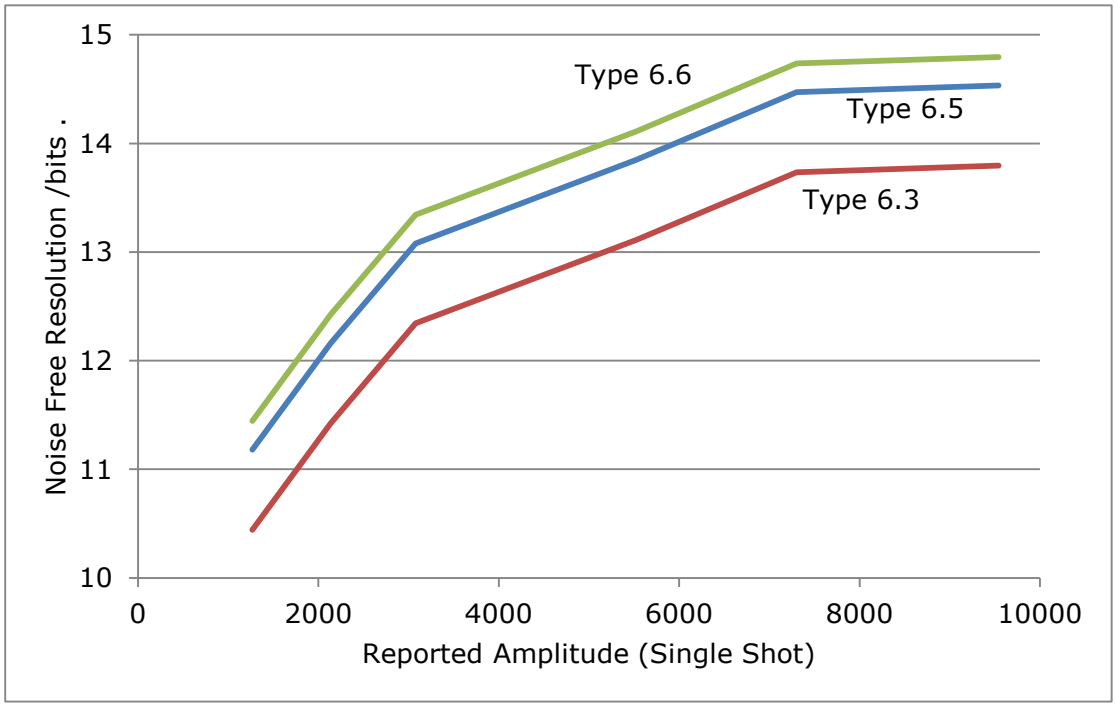


Figure 11 Noise Free Resolution, Type 6 sensors, Single Shot measurement

Pipeline Measurement (section 9.7) is used for higher sample rates than Single Shot. Operating in this mode, Reported Amplitude is approximately 20% higher than Single Shot, while Noise Free Resolution is typically 0.4 bits lower.

Noise Free Resolution can be improved by averaging raw samples from a CTU chip. Averaging 2N samples increases Noise Free Resolution by N/2 bits. So averaging 4 samples (N=2) improves Noise Free Resolution by 1 bit, and averaging 16 (N=4) samples improves Noise Free Resolution by 2 bits. Measurements of Linearity Error and Offset Error are separated from Random Noise by averaging in this way. Noise Free Resolution may also be improved by configuring the CAM502 to digitally filter results, as described in section 9.11.

5 Circuit Layout

This section contains recommendations for the PCB layout of the CAM502 circuitry, and applies to both Type 2 sensor circuitry (section 3) and Type 6 (section 4).

5.1 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 CAM502. Conductors carrying signals not associated with the CAM502 chip should not be placed between the CAM502 chip and ground plane.

Connections from the CAM502's VSS pins to the ground plane must be made with vias placed close to the chip. Capacitors C_VDD, C_AVDD, C_CORE, C_XIN and C_XOUT must be kept close to the chip with short distances to ground connections, as illustrated in Figure 12.

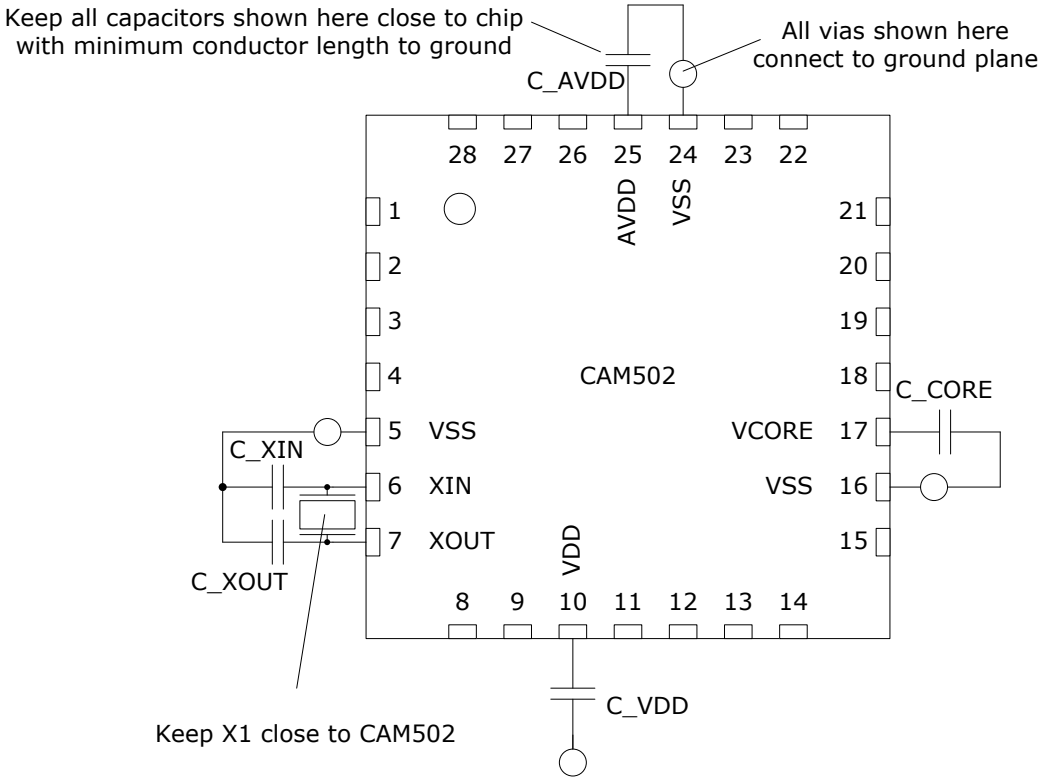


Figure 12 Ground plane vias and capacitors close to the CAM502 chip

It is also important to keep the capacitors used for sensor filtering close to the CAM502 chip, as detailed in the next section.

5.2 Sensor Coil Connections to CAM502

Figure 13 illustrates connections between sensor coils and the CAM502 circuit, including its filter components. When a sensor coil is connected to the CTU circuit, the traces and/or wires forming the connection make a loop. The loop formed by the COSA coil connections is shaded as an example.

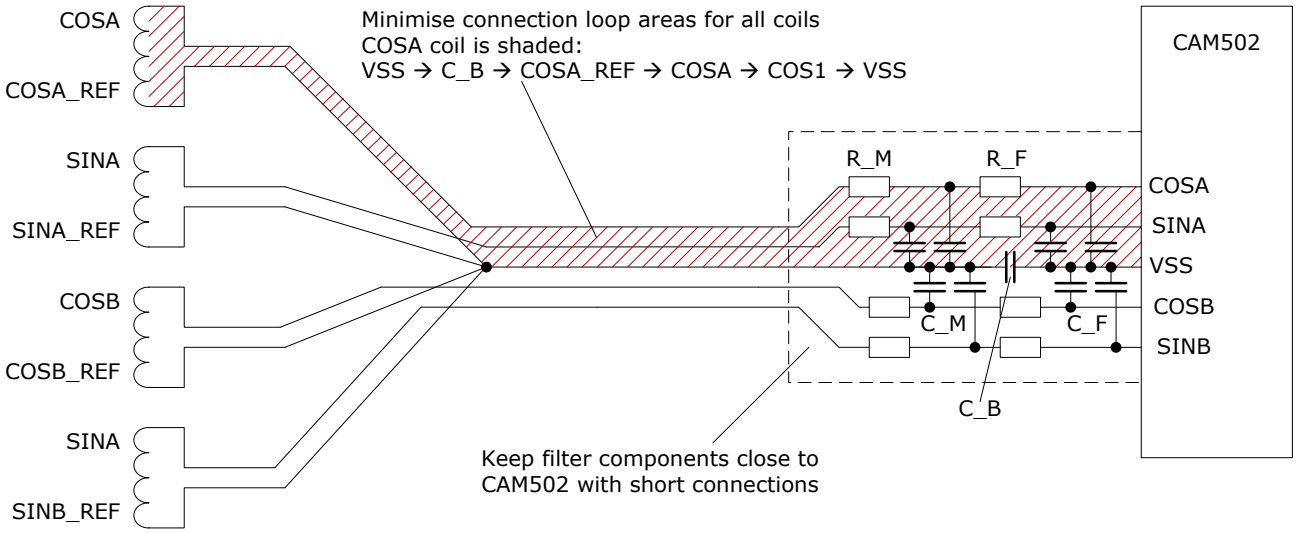
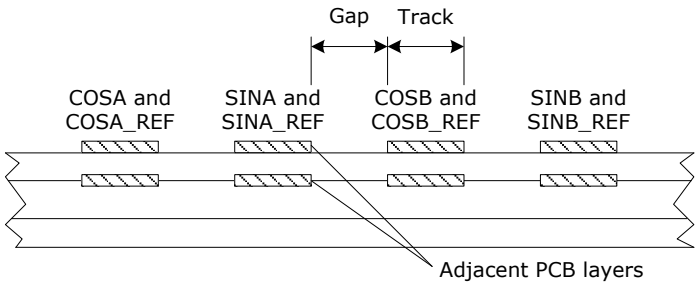
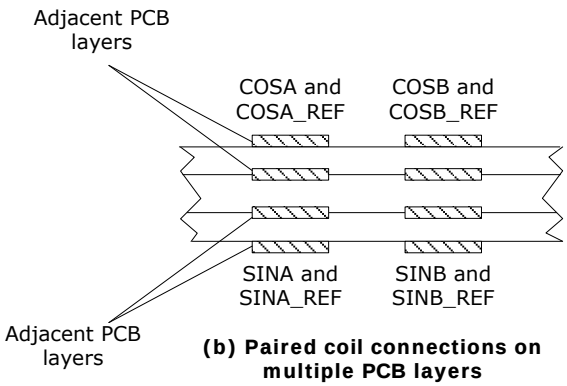


Figure 13 Sensor coil connections to CAM502

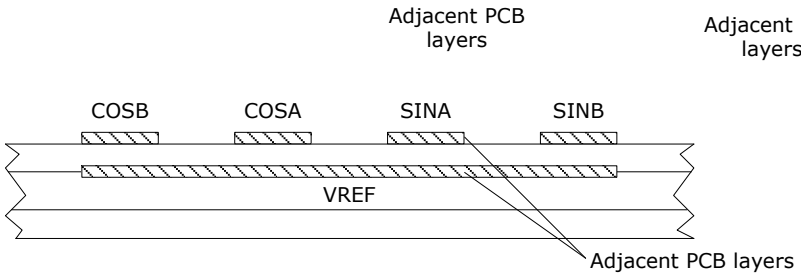
The loop area for each of the sensor coils must be minimised, in order to minimise coupling to the target and/or any AC magnetic interference. Wires used for connection must be run in a tight bundle, on adjacent conductors in a ribbon cable or twisted. Conductors on a PCB should be arranged in coil pairs as in Figure 14 (a) or (b), or adjacent to a common VREF conductor as in Figure 14 (c) or (d).



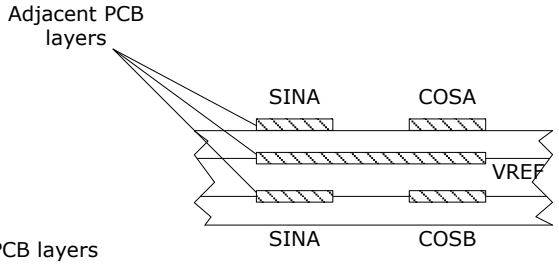
(a) Paired coil connections on 2 PCB layers



(b) Paired coil connections on multiple PCB layers



(c) Common VREF with coil connections on adjacent layer



(d) Common VREF with connections either side

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.

5.3 Excitation Circuit and Coil Connections

The CAM502 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 H-bridge circuit employing two MOSFET pairs (Figure 6), while Type 6 circuitry uses a single MOSFET pair (Figure 8).

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.

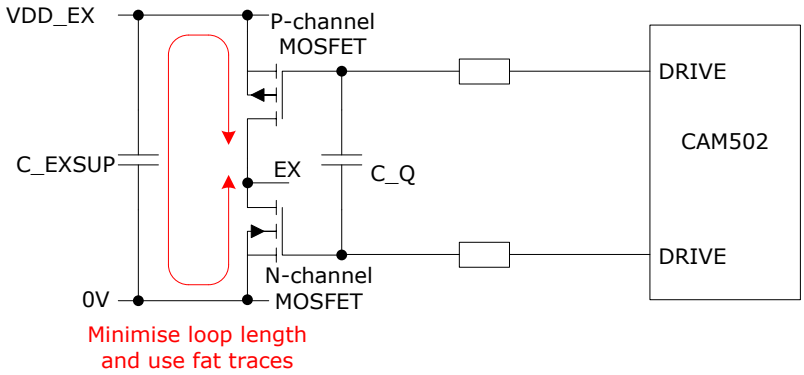


Figure 15 Excitation circuit layout

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 12 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 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 there is a conflict between the two.

Table 12 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

6 Resonator Detection

The CAM502 chip is connected to one or more sensors, and detects the position of inductively coupled resonators inside targets coupled to each sensor. For each measurement it first performs an excitation burst to power the resonator, then switches this off and detects the EMFs induced by the resonator in the senor coils. Please refer to a sensor's datasheet for more details on its principle of operation.

The CAM502 first detects each resonator using a *search* operation with a *broadband* excitation burst to determine its frequency. Subsequent measurements are *narrowband*, designed to maximise resonator EMFs, unless or until the target goes out of range. Each measurement the CAM502 chip updates its estimate of resonator frequency, so that the next excitation burst is at the optimum frequency. That way the CAM502 corrects for changes in resonator frequency due to drift and any changes in metal environment, so that excitation is always at the optimum frequency.

The CAM502 CTU's Nominal Operating Frequency (Table 13) is under crystal control. For diagnostic purposes it reports resonator frequency in the results registers. This Reported Frequency is in Hz relative to the CTU's Nominal Operating Frequency.

Table 13 resonator detection parameters

Item	Min	Typ	Max	Comments
CTU Nominal Operating Frequency	185.99kHz	186.01kHz	186.03kHz	With nominal 8.000MHz crystal assuming ±100ppm across crystals and Operating Temperature
Min resonator frequency for VALID			172.2kHz	
Max resonator frequency for VALID	199.8kHz			
Resolution of Reported Frequency		465Hz		
Resonator Q-factor	60		150	
Minimum Amplitude for VALID	1024		1100	Measured once VALID in single shot mode
Rate of change of resonator frequency			1kHz/ms	For VALID position measurement

The CAM502 detects resonator frequency to yield a VALID position measurement result across a wide range of frequencies, from the minimum to the maximum values shown above. This wide range allows resonators to be manufactured with relaxed frequency tolerance. It also allows for additional margin for positive and negative frequency changes in use, for example due to the presence nearby metals. These allowances are illustrated in Figure 16.

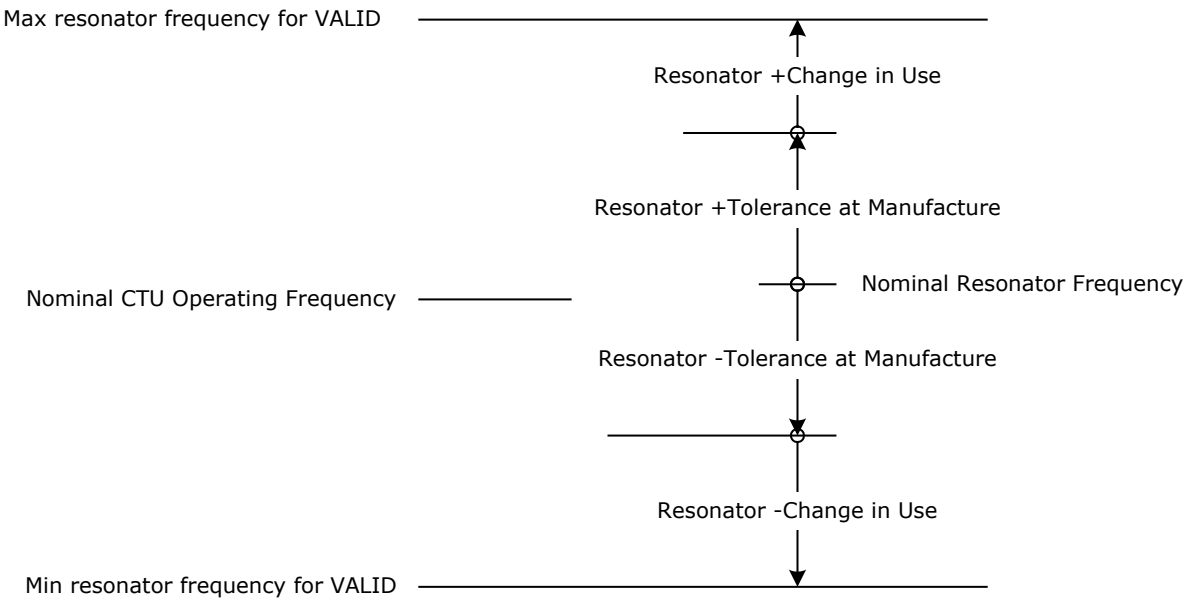


Figure 16 Illustration of frequency tolerance budgets

7 SPI Hardware

7.1 Overview

This section describes how data is written to, and read back from, the CAM502 CTU, over its SPI interface. Each read or read-write operation accesses one or more of the CTU's internal registers.

For communication with a host system, the CAM502 CTU is always an SPI slave. All communication is initiated by the host.

7.2 Data Transfer Method

The CAM502 CTU always operates as an SPI slave device. The host device starts a data transfer by driving nSS low. It sends data to the CAM502 with the MOSI line, and provides the CAM502 with a serial clock line SCK. The CAM502 detects each MOSI bit on the rising edge of SCK. The CTU 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.

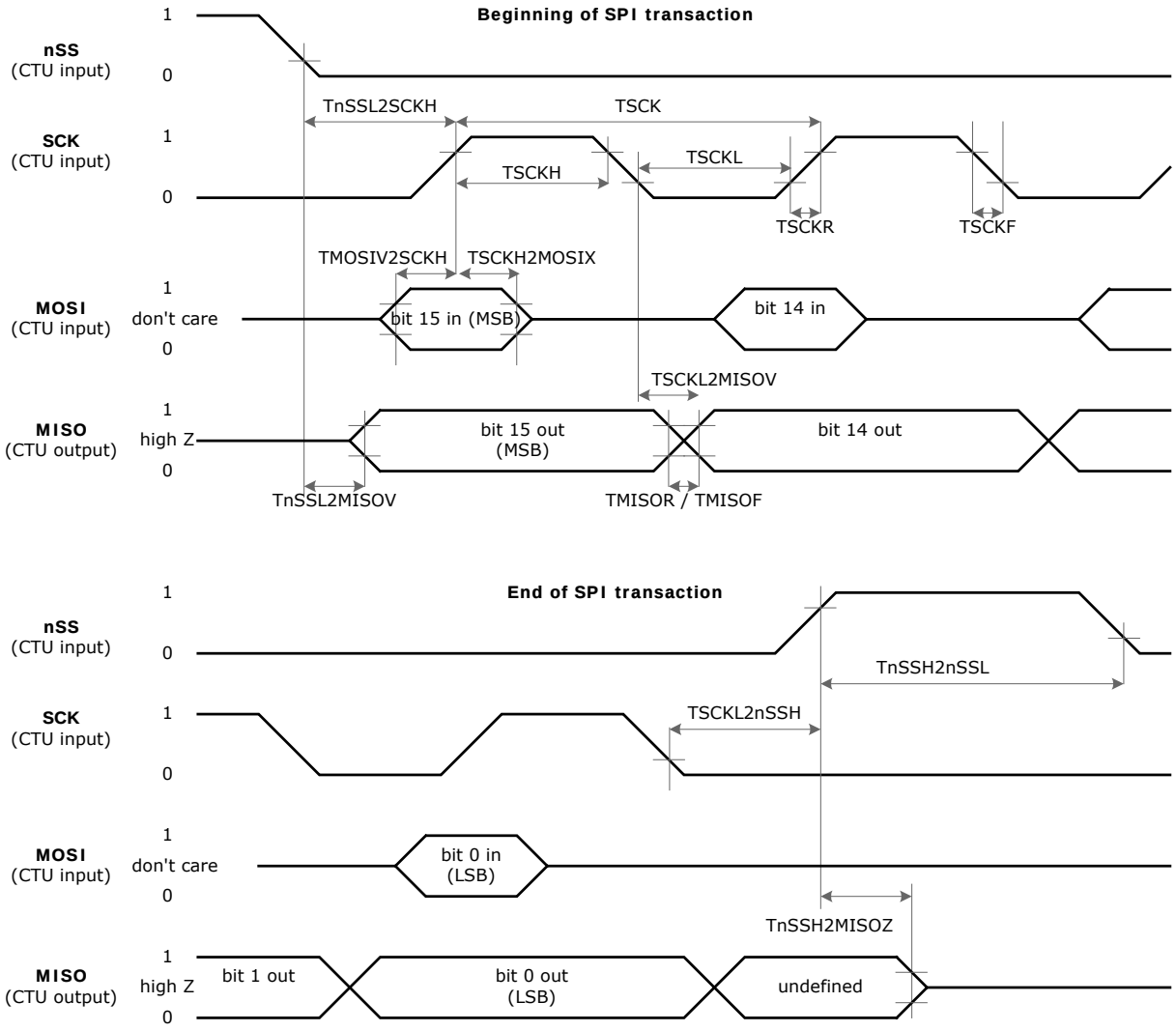


Figure 17 SPI Data Transfer

7.3 Standard SPI Format Register Access

The host operates the CAM502 CTU 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*). The ACS bits also define the format of the next SPI transaction, which may either be *Standard SPI Format* or *Burst SPI Format*. This section describes Standard SPI Format, and section 7.4 describes Burst SPI Format.

Table 14 ACS bits definition

ACS[3:0]	Abbreviation	Access type, this SPI transaction	Format of next SPI transaction
0x0	RS	Read	Standard SPI Format
0x1	RB	Read	Burst SPI Format
0x2…0xD		Reserved, do not use	
0xE	WRB	Write Read	Burst SPI Format
0xF	WRS	Write Read	Standard SPI Format

When the host requests a Read in Standard SPI Format, the next 16 bits clocked out of MISO are the data contained in the register at the specified address. The CAM502 ignores the state of MOSI during data transfer. This transaction is illustrated in Figure 18.

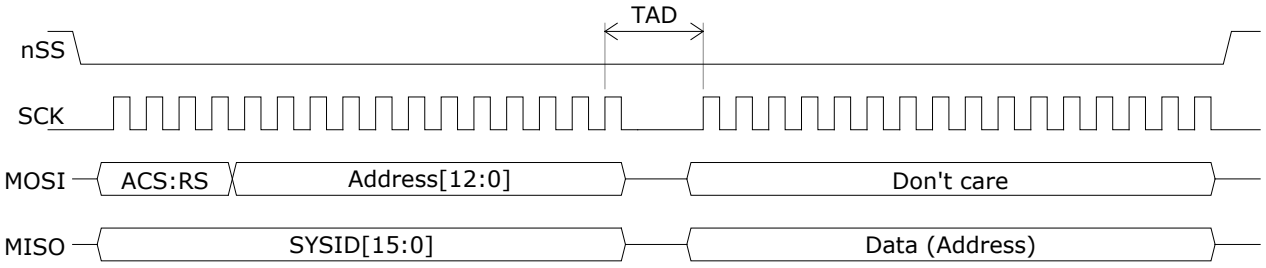


Figure 18 Standard Read, single register

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.

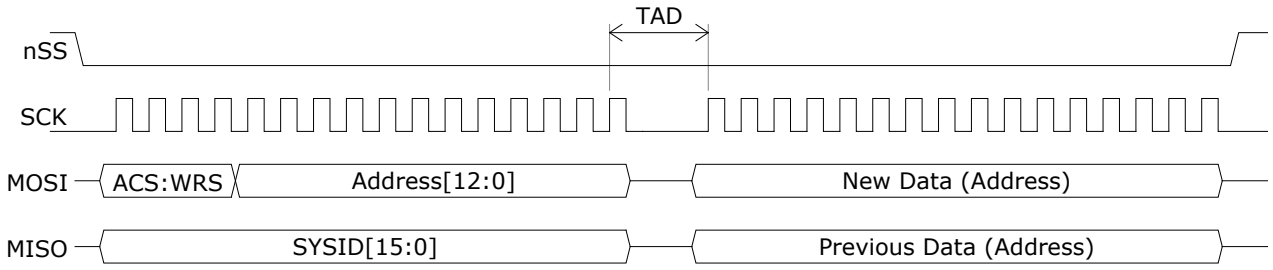


Figure 19 Standard Write Read, single register

For both of these Standard SPI Format 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 CAM502 time to load its SPI transmit buffer with data from the correct address.

The CTU outputs the contents of its SYSID register as the first word of each standard SPI transaction. This defaults to 0xABCD, and may be changed by the host, please see section 8.8. Other values are possible when operating in Bootloader mode, or as a result of errors. Please see Table 27. It is recommended that the host checks the value read back against the expected value as a test for SPI communication integrity.

If the host attempts to access a reserved or unimplemented address, incoming data will be discarded and the state of MISO is undefined.

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 and Figure 21. Data is transferred one register at a time, starting at the first specified address. 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 is not essential to transfer complete 16 bit words into the CAM502. However it is recommended, since data from any incomplete word will be discarded.

There is a minimum time TDD required between adjacent data words, to allow the CAM502 time to load the SPI transmit buffer with data for the next address. This is measured between the last rising edge of SCK of one word and the first rising edge of SCK of the next. At lower bit rates, when TSCK is greater than or equal to TDD, the SCK clock may run continuously with no additional delays. TDD is illustrated in Figure 20 and Figure 21, and specified in Table 15.

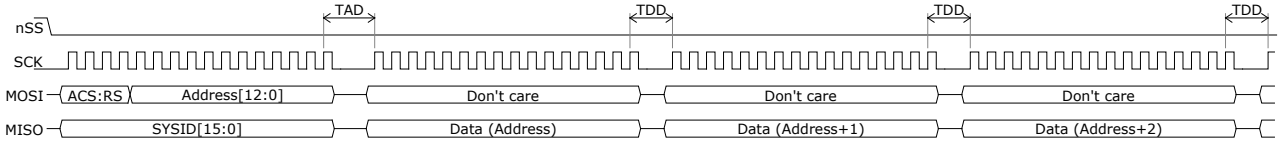


Figure 20 Multiple register access, Read

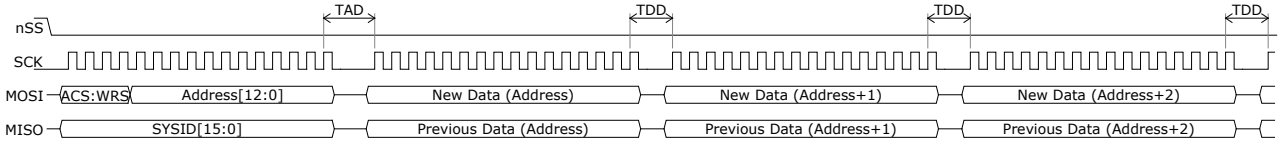


Figure 21 Multiple register access, Write Read

7.4 Burst SPI Format Register Access

The host may request that the next SPI transaction has Burst SPI Format using the appropriate ACS bit settings listed in Table 14. Burst SPI Format has the following features:

- Data is read from the address specified by the host in the *previous* SPI transaction.
- Data is written to the address specified by the host in the *current* SPI transaction, like Standard SPI Format.
- The host transmits data immediately, and does not prefix data with the SYSID register.
- The time delay TAD is not required between Address and Data, because the CAM502 loads its SPI buffer before the next SPI transaction. The shorter time TDD is required between all words instead.
- The maximum value of TnSSL2MISOV and the minimum value of TnSSL2SCKH are increased slightly to allow the CAM502 time to set up the first SPI word to transmit to the host (Table 15). The increase is small compared to the overall decrease in SPI transaction time due to the elimination of TAD, so that SPI transaction times are reduced significantly. Please see Table 16 for examples.

These features are intended to speed up data access for systems requiring low latency operation, particularly when repeated and rapid access to a sensor's results registers is required.

Figure 22 and Figure 23 both illustrate SPI transactions with Burst SPI Format; the previous ACS bits must have been either RB or WRB. The first word out of MISO is data from the register location whose address was specified in this previous SPI transaction, and it is followed by data from the following addresses.

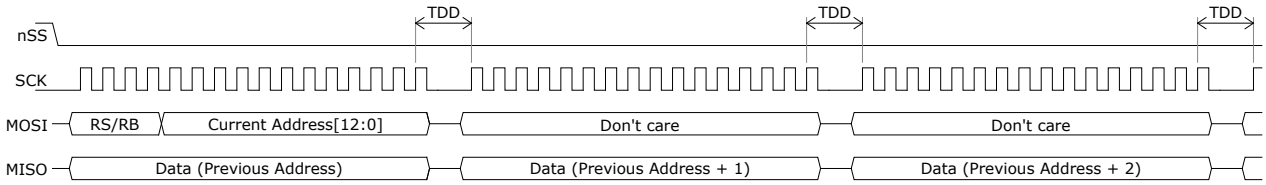


Figure 22 Burst SPI Format, Read

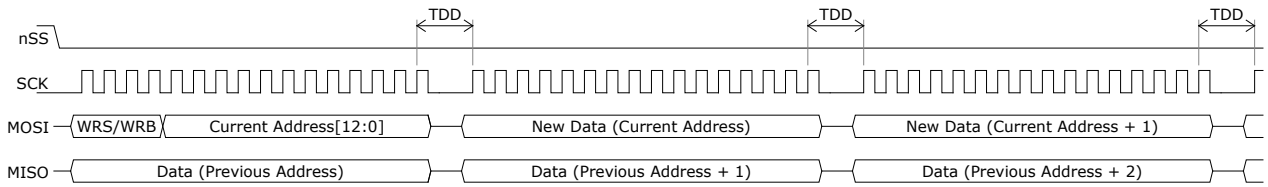


Figure 23 Burst SPI Format, Write Read

Figure 22 illustrates the case when the current SPI transaction is a Read. The ACS bits can be set to either RS (Read, next SPI transaction Standard SPI Format) or RB (Read, next SPI transaction also Burst SPI Format).

Figure 23 illustrates the case when the current SPI transaction is a Write Read. It can either be a WRS (Write Read, next SPI transaction Standard SPI Format) or WRB (Write Read, next SPI transaction also Burst SPI Format). In both cases, the host should transmit data after the address bits. This is written to the register address specified in the current SPI transaction. Note that a Burst SPI Format Write Read allows the host to read from one more register than it writes to.

An additional feature of Burst SPI Format is that is allows the host to Read CAM502 registers with a single 16-bit word SPI transaction, as illustrated in Figure 24.

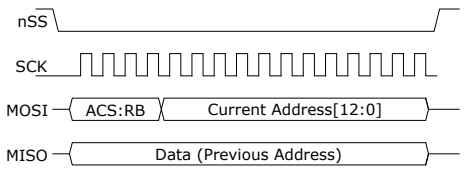


Figure 24 Burst SPI Format, Read from one register

7.5 Interface Timing Specifications

Table 15 interface timing specifications

Parameter	Description		Min	Typ	Max	Units
TVALIDCHECK	Time for CTU internal validity checks following reset		5	-	12	ms
TSCKL	SCK Input Low Time		30	-	-	ns
TSCKH	SCK Input High Time		30	-	-	ns
TSCK	SCK clock period		100	-	-	ns
TSCKR	SCK Input Rise Time		-	10	25	ns
TSCKF	SCK Input Fall Time		-	10	25	ns
TMISOR	MISO Rise Time (50pF load)		-	-	25	ns
TMISOF	MISO Fall Time (50pF load)		-	-	25	ns
TMOSIV2SCKH	MOSI Setup Time		30	-	-	ns
TSCKH2MOSIX	MOSI Hold Time		30	-	-	ns
TSCKL2MISOV	MISO valid after SCK edge		-	-	30	ns
TnSSL2MISOV	MISO valid after nSS edge	Standard SPI Format	-	-	100	ns
		Burst SPI Format	-	-	230	ns
TnSSL2SCKH	nSS low to SCK edge	Standard SPI Format	120	-	-	ns
		Burst SPI Format	250	-	-	ns
TSCKL2nSSH	Last SCK edge to nSS high		100	-	-	ns
TnSSH2MISOZ	nSS high to MISO high Z		10	-	50	ns
TAD	Address end to first data, Standard SPI Format only	CONT=0	5	-	-	µs
		CONT=1	7	-	-	µs
TDD	Delay between data words		250	-	-	ns
TnSSH2nSSL	nSS high time following read of register(s), excluding pipeline measurement		6	-	-	µs
	nSS high time following write to register(s) controlling IO, LED or DAC configuration		200	-	-	µs
	nSS high time during pipeline measurement		174			µs
	nSS high time, any other SPI transaction		40	-	-	µs
TnSSH2IOch	Time to clear an IO by clearing SIF or PTF bits		-	-	40	µs
	Time to change an IO following write to register(s) controlling IO, LED or DAC configuration		-	-	200	µs

7.6 Minimum SPI Transaction Times

The time taken for any SPI transaction depends on the actual host timings used and the number of registers accessed. Table 16 provides examples of the minimum SPI transaction time, measured from nSS low edge to high, and based on minimum interface timings taken from Table 15.

When operating in Continuous mode with CONT=1, and using standard SPI transactions, a larger value of TAD is required, as described in section 9.10.

Table 16 minimum SPI transaction times

Registers accessed	Standard SPI Format, CONT=0	Standard SPI Format, CONT=1	Burst SPI Format
SCW, PTEF, RESA…RESE	20.5µs	22.5µs	15.9µs
RESA, RESB, RESC	12.0µs	14.0µs	5.4µs

8 Register Description

The host configures and controls the CAM502 CTU 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 CTU's registers.

Reading and writing to registers is done over the SPI interface, using the procedures described in section 7. Section 9 describes how to control the CTU to perform sensor position measurements, and how to configure the function of the CTU's IO pins.

The CTU'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. The register map shows a number of registers whose functions are not implemented in the CAM502, 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 CAM502.

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	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	TPT1E	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

	Address	Description	Register	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
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	(Reserved)																	
	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

Read Only | Read / Write | Read / Write to 0

Figure 25 Register map

8.1 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. SPI transactions will be ignored until the device comes out of reset.

The PWRDN bit is not currently implemented.

Writing a 1 to the BOOTLOAD bit resets the CTU at the end of the SPI transaction. When the CTU comes out of reset, it will remain in Bootloader Mode (section 10) until the next reset.

Writing 0x0C1C to the SAVEKEY register then 1 to the SAVE bit will result in the current register contents being made the Configurable Defaults. Writing 0x0C1C to the SAVEKEY register then 1 to the FACTORY bit, will result in restoration of the factory default register values. After SAVE or FACTORY operations the CTU will reset. Please see section 9.13 for more details.

8.2 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

In continuous mode (CONT=1), SYSI determines the interval between measurements, and is timed by the CTU. In Continuous Pipeline Mode with PIE=1 it is in multiples of 0.1µs, otherwise with PIE=0 it is in multiples of 100µs.

In Pipeline Mode with CONT=0, SYSI is used to tell the CTU what sample interval the host is using between SPI transactions, so that it can ensure fresh measurement results are available just in time for the host to read them. Writing 0 to SYSI tells the CTU to use the interval between the last two measurements as the estimate of the next sample interval (*CTU Timed Pipeline Interval)*. Writing a non-zero value to SYSI tells the CTU to use that value of SYSI as the sample interval instead (*Host Timed Pipeline Interval*). Please refer to section 9.7 for details. In both cases the interval is measured in multiples of 0.1µs.

8.3 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 17. Independently, each pin may be configured for open drain or *digital* with the coding of Table 18. *Digital* means driven both high and low.

Table 17

INTnAH	Function
0	Active Low
1	Active High

Table 18

INTnDO	Function
0	Open Drain
1	Digital Output

8.4 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 = 0x0000

This system register is not currently implemented.

8.5 SAVEKEY: Save Key

BOOTVER	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. Both of these operations result in an internal reset, which clears SAVEKEY back to its default, "safe", state.

8.6 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 = 0x????

The BOOT register contains the fixed revision number for the CTU's Bootloader Code, and is read-only.

8.7 SYSVER: System Version

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 = 0x????

The SYSVER register contains the version number of the CTU's Application Code, and is read-only. It is updated when new code is successfully updated using the bootloader (section 10).

8.8 SYSID: System Device ID

SYSID	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00F	DEVICEID[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 in normal operation (see Figure 19). It is recommended that the host verifies this value against the expected one, to assist detection of communication errors.

The Device ID 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 CTU. It is recommended to avoid the codes listed in Table 27, since these are used to report status.

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

8.9 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	-	TRIG	CONT	GO
Access	R				R/W	R/W0	R/W	R	R/W	R/W0	R/W0	R	R	R/W	R/W	R/W

Factory default value = 0xn000

The registers described above relate to system functions. The Sensor Control Word, and the other *sensor registers* described below, relate to sensors connected to the CTU. The CAM502 only supports one connected sensor, and the host must only access sensor 1 (SENSOR=1).

A sensor's SCW register controls the CTU's measurements of that sensor.

The upper 4 bits of SCW, SENSOR[3:0], is a read-only field that equals the sensor number, to help the host verify it associates results with the correct sensor.

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 CTU will measure that sensor continuously. In this case, the sample interval between measurements is configured with SYSI, see section 8.2. If a sensor's CONT bit is set to 0, setting its GO bit to 1 will result in a single shot measurement. The CTU will clear GO to 0 upon completion.

The TRIG bit is not implemented.

The VALID bit indicates that the last measurement result for the sensor was valid. The CTU 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 the limits specified in Table 13.

The NEW flag indicates that new measurement data is available for the sensor. The CTU sets NEW to 1 when a measurement is completed. If the host uses a write read SPI operation to read results, spanning registers SCW and the desired result registers, it is recommended to clear NEW back to 0 during that transaction. That way, the host can unambiguously verify that the results it has collected from the CTU 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. The SIE bit controls whether sample indicators are enabled. Once activated, the host may clear a sample interrupt by writing a 0 to SIF.

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 9.8 for details.

PIE and PIF are for operation in Pipeline Mode (section 9.7). PIE enables Pipeline Mode. PIF is a flag that indicates when the sensor is being sampled in Pipeline Mode.

8.10 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 most recent measurement. The CTU does not update the SCW and results registers during SPI transactions, to prevent data skew. So their values are effectively latched by the negative going transition of nSS.

The CTU supports different sensor types. The interpretation of result registers depends on the sensor type as detailed in Table 19.

Table 19 Results register contents by Sensor Type

Register	Sensor Type
	2 and 6
RESA	CtuReportedPositionI32 high word
RESB	CtuReportedPositionI32 low word
RESC	AmplitudeAU16
RESD	AmplitudeBU16
RESE	RelativeFrequencyI16
RESF	BAPositionMismatchI16

CtuReportedPosition may be converted into measurement units with Equation 3 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 (SinLengthA) for use in Equation 3. For full 360° rotary Type 6 sensors, SinLength equals 360°/Subtype.

$$Reported\ Position = \frac{CtuReportedPosition}{65536} \times SinLength$$

Equation 3

Note that Reported Position is a signed quantity which is nominally 0 when the Target Origin/ Reference Direction is aligned with the Sensor Origin/Reference Direction for linear/rotary sensors. 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 (for example see section 4.4). Type 2 and 6 sensors have two Amplitude results, one derived from measurements of the fine sensor coils, *Amplitude A*, and one from the coarse, *Amplitude B*. 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 CTU's measurement of the frequency difference in Hz between the target and the CTU's Nominal Operating Frequency listed in Table 13.

BA Position Mismatch is specific to Type 2 and 6 sensors. It is scaled in the same way as Reported Position using Equation 3. 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 9.8).

8.11 CONST: Measurement Constants

CONST	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xn08	-								FCSEL				ABSSEL			
Access	R								R/W				R/W			

Factory default value = 0x000C

The CONST register includes the measurement constants FCSEL and ABSSEL.

ABSSEL may be used to specify a number of fine measurements, ABSCOUNT, to be taken in between single coarse measurements when operating in Pipeline mode, see section 9.9. The value of ABSCOUNT is given by:

$$ABSCOUNT = 2^{ABSSEL}$$

Equation 4

FCSEL controls whether or not the CAM502 applies digital filtering to position results, and the amount of filtering, as described in section 9.11.

8.12 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 = 0x0002

The CTU supports different Types of sensor (Table 1). Only one Type of sensor may be connected to the CTU. Sensor 1's TYPE field is used to tell the CTU what sensor Type is connected, including when there are multiple sensors.

Some sensors have Subtypes. For example "Type 6.5" is Type 6, Subtype 5. For sensor Types that support Subtypes, the sensor's Subtype must be written to the SUBTYPE field. If the sensor Type also supports multiple sensors, SUBTYPEs may be different for each sensor. When the sensor Type does not support Subtypes, SUBTYPE must be set to 0 for each sensor used.

8.13 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 CTU has new, or new valid, measurement results available for a sensor. The function is configured with SCTRL[1:0], see Table 21.

Sample indicators can be configured to control the CTU's digital IOs. SMAP[3:0] configures which of the available IOs is used for the sensor's sample indicator. See Table 20 for how this bit field is encoded.

Sample indicators can be reset by the host over the SPI interface by writing a 0 to the sample indicator flag SIF, see section 8.9. Alternatively, they may be auto-cleared. SAUTOCLR[1:0] controls sample indicator auto-clear behaviour. Table 22 defines how this bit field is encoded.

The *clear before measurement* setting ensures that subsequent activation of the sample indicator causes an edge on the appropriate IO, for example every time new measurement results are available when SCTRL is 1.

The *clear after measurement* setting avoids edges unless there is a change in state, for example between VALID and NOT VALID when SCTRL is set to 2.

Table 20

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

Table 21

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

Table 22

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

9 System Operation and Measurement Timing

9.1 Reset, Validity Check and Reading Versions

The CTU is internally reset following power on, when nRESET is toggled or when the host writes a 1 to the RESET, SAVE or FACTORY bits in the SYSCW register (section 8.1).

Following a reset, the CTU performs internal validity checks, as illustrated in the flowchart of Figure 26.

Validity checks include a checksum of the CTU's FLASH memory contents, including Application Code. The time taken for these checks is TVALIDCHECK, as specified in Table 15. For normal operation, the host should allow at least TVALIDCHECK(MAX) between a reset and the first SPI communication with the CTU, to avoid interrupting the validity checks.

The CTU has a set of 4 registers that contain version numbers, from CTUID (address 0x000B, see section 8.4) to SYSVER (0x000E, section 8.7). It is recommended that the host reads these 4 registers as the first SPI transaction after any reset. If validity checks passed successfully, the CTU will return the SYSID register contents (factory default 0xABCD) as the first word, followed by the version numbers.

If the CTU 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 10 for details of loading code using the CTU's bootloader.

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

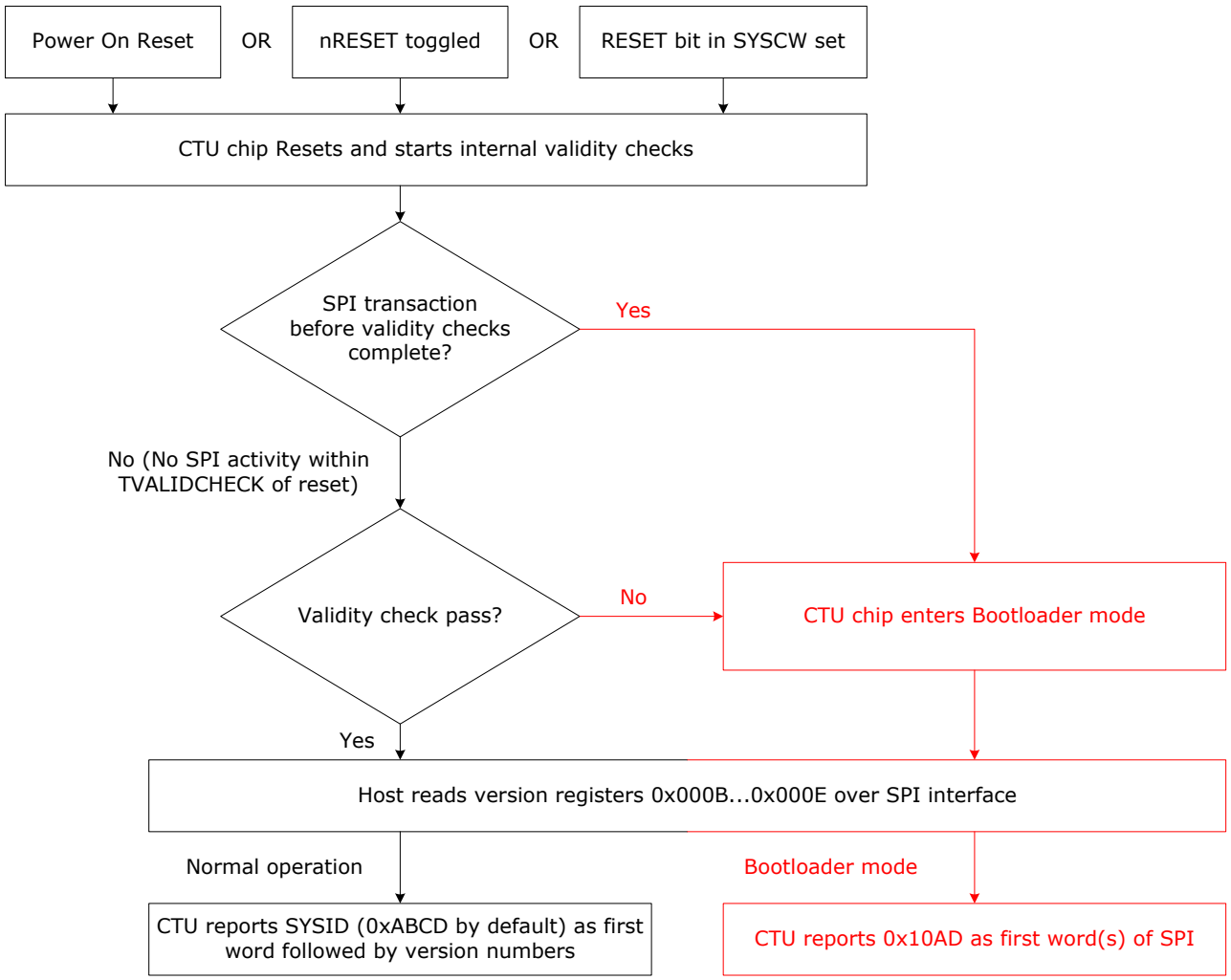


Figure 26 CTU reset and validity checking process

9.2 Setting Sensor Type and Subtype

When connected to a Type 2 sensor, set sensor 1's STYPE:TYPE to 2 and STYPE:SUBTYPE to 0.

When connected to a Type 6 sensor, set sensor 1's STYPE to 6 and SUBTYPE to match the sensor's SUBTYPE. For example a Type 6.5 sensor has Subtype 5.

Section 8.12 has more details of the STYPE register.

9.3 Sample Indicators for Signaling Measurement Completion

The CTU's IOs may be configured to indicate when a new sample is available after a measurement is complete on any sensor. This is typically used to signal to a host device when to read out measurement results. Table 23 lists the register settings required.

Table 23 Configuring Sample Indicators for signaling measurement completion on an IO

Register : bits	Controls…	Reference
SYSIO: INTnAH	Whether the selected IO ("n") is active low or active high	Section 8.3
SYSIO: INTnDO	Whether the selected IO ("n") is open drain or digital	
SIC:SMAP	Which of the CTU's IOs is selected for the sensor's sample indicators	Section 8.13
SIC:SCTRL	Whether the IO activates every measurement, or only when its result was VALID	
SIC:SAUTOCLR	Automatic clearing of the sample indicator. Set to *OFF* if the host uses write read SPI transactions for reading results. When reading results, the host should write 0 to SCW:SIF to clear the sample indicator again ready for the next measurement. Set to *clear before measurement* if the host uses read-only transactions for reading results. This will ensure there is an inactive to active edge on the selected IO to signal the measurement is complete.	
SCW:SIE	Whether sample indicators are active. Set to 1 when writing to SCW to start measurements.	Section 8.9

9.4 Sample Indicators for Generating "VALID" Output

As an alternative to signaling measurement completion, Sample Indicators may be used to signal whether the most recent measurement on any sensor was VALID or NOT VALID. Table 24 lists the register settings required.

Table 24 Configuring sample indicators to generate a VALID output on an IO

Register : bits	Controls…	Reference
SYSIO: INTnAH	Whether the selected IO ("n") is active low or active high	Section 8.3
SYSIO: INTnDO	Whether the selected IO ("n") is open drain or digital	
SIC:SMAP	Which of the CTU's IOs is selected for the sensor's sample indicators	Section 8.13
SIC:SCTRL	Whether the IO activates every measurement, or only when its result was VALID. Set to *if VALID flag set* so that the IO reflects VALID.	
SIC:SAUTOCLR	Automatic clearing of the sample indicator. Set to *clear after measurement* so that the IO remains active until a measurement result is NOT VALID.	
SCW:SIE	Whether sample indicators are active. Set to 1 when writing to SCW start measurements.	Section 8.9

Note that the IO will reflect the result of the most recent measurement, and will not change state unless measurements are taking place.

9.5 Single Shot Measurement

To perform a single measurement from one sensor, the host should write the GO bit in the sensor's SCW register to 1. In response, the CTU performs a position measurement on that sensor. The position measurement comprises a "pulse" during which the resonator is excited, followed by an "echo" when its EMFs are detected in sensor coils. The CTU then performs a position calculation. The process is illustrated in Figure 29.

The selected IO will activate upon completion if sample indicators are appropriately configured (section 9.3). The host can use this edge as the trigger to perform an SPI transaction that reads measurement results. For the edge to occur correctly, the initial SPI transaction setting GO=1 should also clear sample indicators (SIF=0).

To read measurement results, it is recommended that the host performs a single Write Read transaction that spans the SCW register and the results registers of interest. The beginning of such a transaction is illustrated in Figure 27. The first 4 bits transmitted on MOSI should be the ACS code for Write Read, followed by the 4-bit sensor number then 0x00 so that the sensor's SCW register will be accessed first. Please refer to section 7.3 for more details on register access.

In the SCW register to write, GO should be cleared (GO=0) to avoid starting another measurement. NEW should also be cleared too (NEW=0), so that the value of NEW read back from the CTU indicates whether the measurement results are old (NEW=0) or new (NEW=1). Sample indicators should be kept active if used (SIE=1).

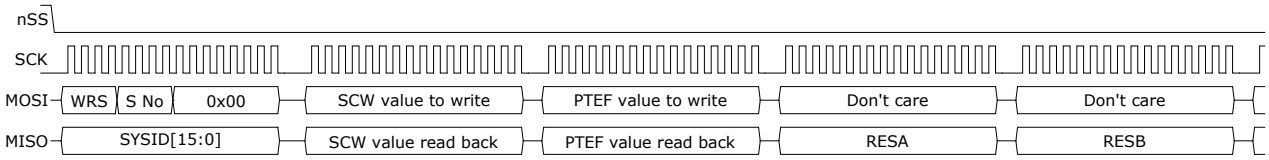


Figure 27 Write Read SPI transaction accessing SCW and measurement results, Standard SPI Format

The host may alternatively use Burst SPI Format to access the SCW and results registers, to reduce transaction time. This SPI format is described in section 7.4, and the beginning of an SPI transaction is illustrated in Figure 28.

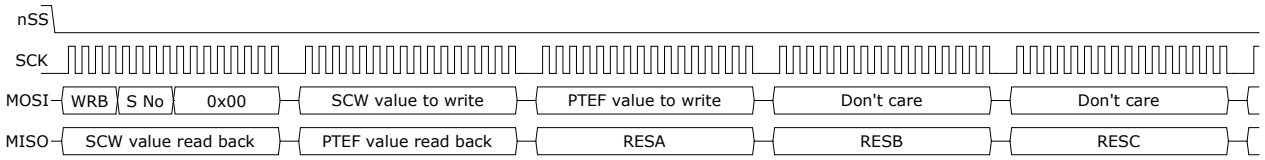


Figure 28 Write Read SPI transaction accessing SCW and measurement results, Burst SPI Format

The host does not need to use sample indicators. It may instead repeat SPI write reads until the measurement has completed (the reading from the NEW bit is 1), and/or allow sufficient time for the measurement to complete.

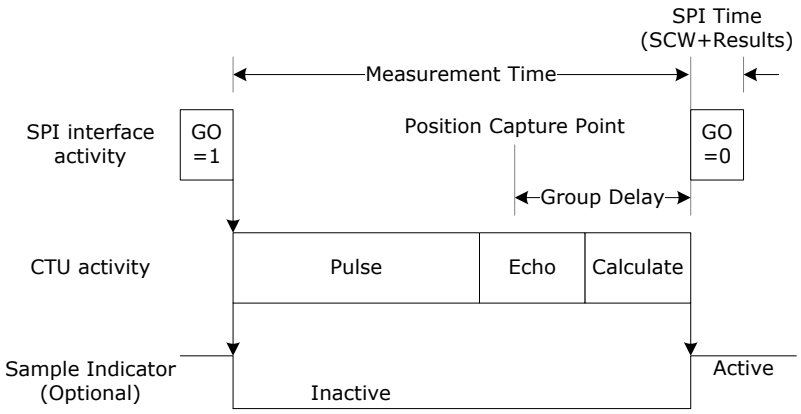


Figure 29 one single shot measurement (if Type 2 and 6, INCF=1)

Figure 29 illustrates a measurement using a single pulse echo. For sensor Types that include both fine and coarse sensor coils, this type of measurement is performed in Incremental mode (INCF=1). An Absolute mode measurement is performed when no target has yet been detected (previous measurement VALID=0), and/or when requested by the host (INCE=0). This comprises measurements from both fine and coarse coils, and the single shot measurement

process is illustrated in Figure 30. Incremental and Absolute mode operation is explained in 9.8, and Table 26 specifies Measurement Time for each mode.

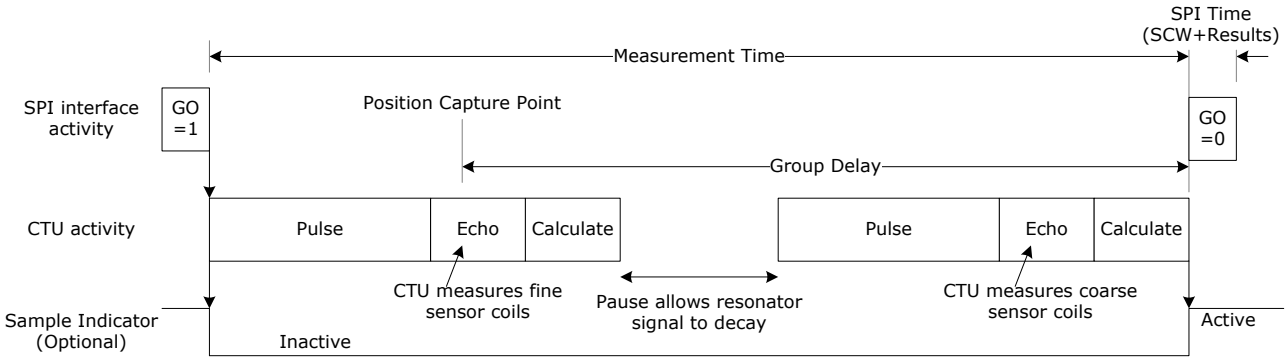


Figure 30 one single shot measurement, Type 2 or 6 with INCF=0

The CAM502 chip's sampling process captures average target EMFs during the echo period, as the resonator is ringing down. It therefore captures the average position, weighted according to the decaying resonator signal amplitude. The actual Position Capture Point comes slightly before the middle of the echo. Group Delay is defined as the time between this Position Capture Point and the CTU ready to provide VALID position data for the measurement.

Additional time is required for the host to read out this data. This minimum SPI Time depends on the number of registers the host reads, and the SPI timings. For convenience, Table 16 includes the minimum time for critical SPI transactions including results access.

9.6 Repeated Single Shot Measurement

The host may perform repeated single shot measurements. A minimum Sample Interval is required to allow the resonator signal to decay sufficiently between measurements, specified in Table 26. As with Measurement Time defined in the previous section, Sample Interval is longer for absolute measurements (INCF=0) than incremental (INCF=1). The figures in this section illustrate incremental measurements, but the process is the same for absolute.

The process of Figure 29 may be repeated, as in Figure 31. This method yields the minimum delay between measurement and the host having its results, but requires two SPI transactions per measurement.

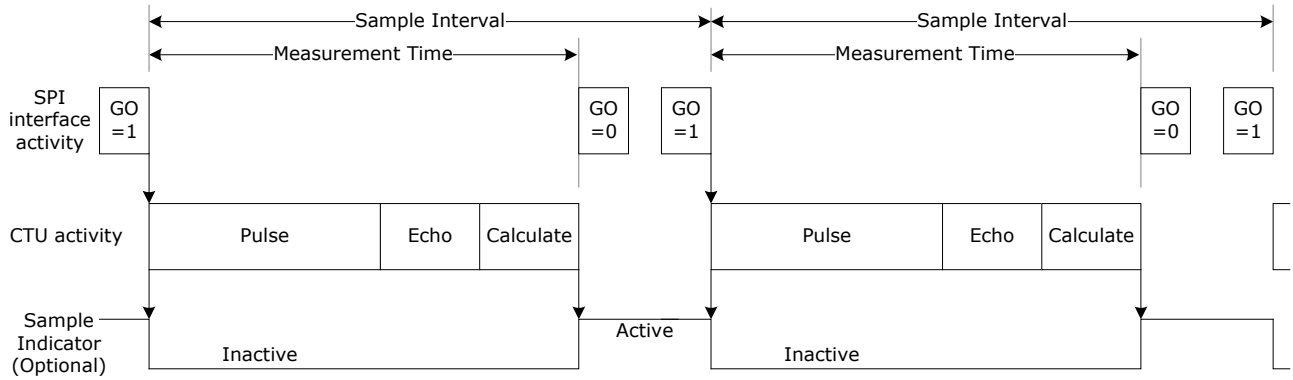


Figure 31 repeated single shot measurement, minimum delay

Alternatively, a single SPI transaction may be used to both read results and start the next measurement, as illustrated in Figure 32. The delay between measurement and the host having results is greater. This approach may be used without sample indicators, providing the time between adjacent SPI transactions exceeds the CTU's maximum Sample Interval.

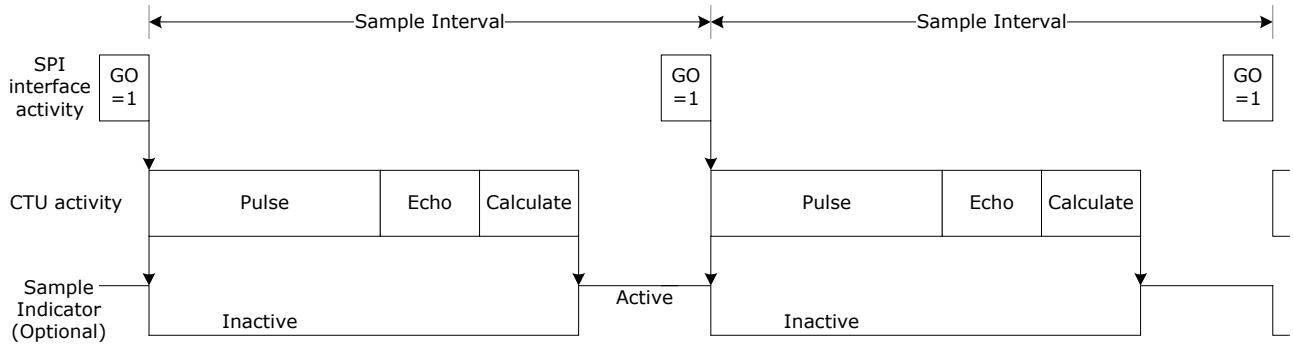


Figure 32 repeated single shot measurement, minimum SPI transactions

It is also possible for the host to set GO=1 in an SPI transaction performed immediately after measurement completion, as in Figure 33. The CTU will start the next measurement as soon as it is able afterwards. This approach yields both a minimum delay and maximum rate, but requires the host to time its SPI transactions when sample indicators are activated.

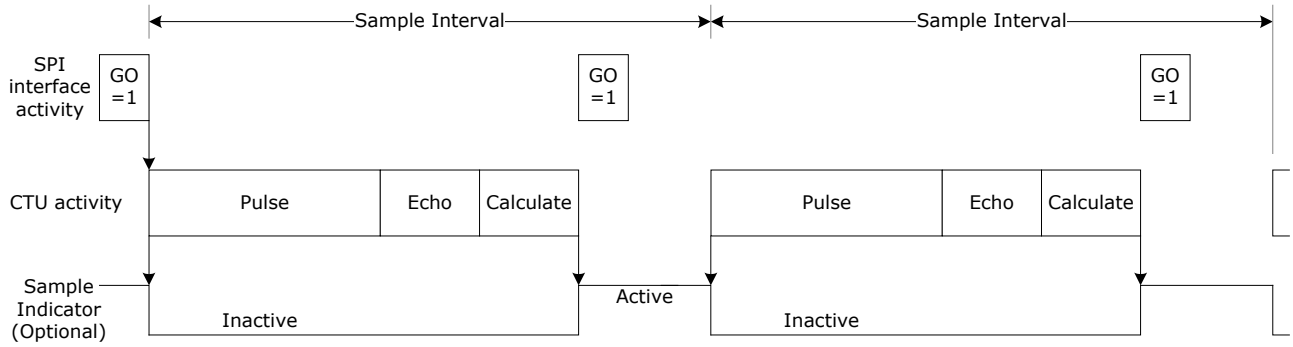


Figure 33 repeated single shot measurement, maximum rate

For even higher sample rates it is possible to use Pipeline Measurement, as outlined below.

9.7 Pipeline Measurement

Pipeline Measurement yields the highest possible sample rate. To reduce the time between measurements, the CTU performs calculation in parallel with the next measurement's excitation *Pulse*, as illustrated in Figure 34. The CTU also adjusts the timing of the excitation waveform so that it continuously reinforces the resonator's signal, rather than allowing it to decay.

Pipeline measurement is activated by setting Pipeline Enable to 1 (PIE=1). The PIE bit is in the SCW register, together with the GO bit. The CTU's nCTRL input is used to synchronise the CTU to the host. nCTRL must be connected to nSS.

The CTU captures the data it needs to determine position throughout the *Echo* period illustrated in Figure 34. The *Position Capture Point* is the effective point in time at which the CTU measures position. The CTU calculates position from the captured data during and after the Echo period. When the calculation is complete, the host may read out the data. The CTU's Group Delay is measured from the Position Capture Point to the negative going edge of nSS for the SPI transaction used to read out the calculation results.

There must be some time period, *TResults2nCTRL*, between the calculation ending and the nSS edge used to read out the result, otherwise the host will read out old data. This time period contributes to Group Delay, so the CAM502 attempts to minimise it. Since the host is in control of when the negative edge of nSS occurs and not the CTU, it is important for the CTU and host to synchronise their activities in pipeline mode.

Most importantly, pipeline mode is designed for constant or near constant Pipeline Interval. When the value of Pipeline Interval is known, the CTU adjusts the time from the negative going edge of nCTRL to the end of the excitation *Pulse*, *TnCTRL2Echo*, so that there is just enough time to complete the Echo and calculation before the next negative going edge of nCTRL.

There are two methods for the host to communicate the expected value of Pipeline Interval to the CTU: *CTU Timed Pipeline Interval* or *Host Timed Pipeline Interval*. To select CTU Timed Pipeline Interval the host should write 0 to the SYSI register (section 8.2). In this case, the CTU measures Pipeline Interval itself, and adjusts its timing on the

assumption that the current Pipeline Interval will be the same as the previous. Host Timed Pipeline Interval allows the host to define the expected time between negative nCTRL edges, by writing the value (in multiples of 0.1µs) to the SYSI register. The value must lie within the allowed range of Pipeline Interval (Table 26).

If CTU Timed Pipeline Interval is selected, the difference between successive Pipeline Intervals must be less than the minimum TResults2nCTRL specified in Table 26, otherwise the CTU may not have new data available in time.

With Host Timed Pipeline Interval selected, the host can write the minimum expected value of Pipeline Interval to SYSI, including any timing jitter that is expected. This mode of operation is therefore more flexible, but requires the host to take greater responsibility for timing.

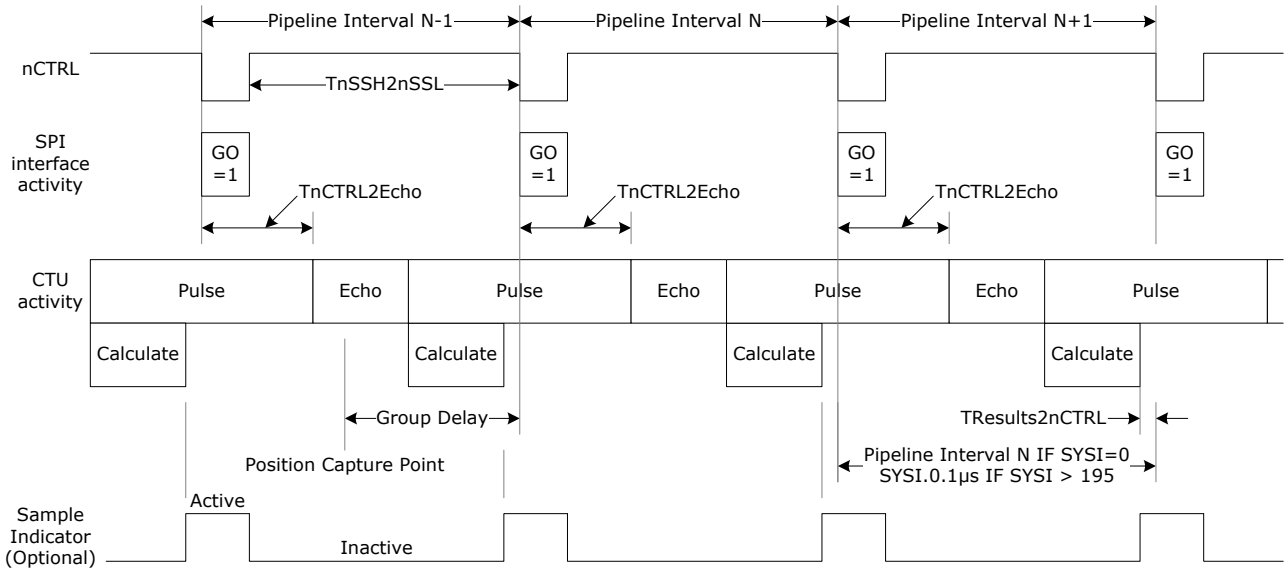


Figure 34 pipeline measurement

The host may begin sampling in pipeline measurement mode (PIE=1) at any time, and may immediately operate at the required Pipeline Interval.

The sensor's SCW register includes a flag to indicate that pipeline measurement is active (PIF=1).

The host must use a write read SPI transaction when configuring the SCW register (PIE=1, GO=1, INCE=1 etc). Subsequent transactions may be Read Only. Either Standard or Burst Mode SPI Formats may be used, as detailed in section 7.3 and section 7.4 respectively.

The time between SPI Measurements, Tn22L2nSSH shown in Figure 34, must be greater than or equal to the minimum specified in Table 15. This means that short SPI transactions are required when the Pipeline Interval is short.

It is recommended to read from the SCW register and all results registers applicable to the sensor's Type for full system health monitoring. However in circumstances where speed is of the essence the read operation may span a smaller number of registers. The example of Figure 35 uses Burst SPI Format to read from only the RESA, RESB and RESC registers. This allows the host to read position (RESA and RESB) and AmplitudeA (RESC), which are the most important data. An Amplitude reading less than the minimum for VALID (Table 13) may be interpreted as VALID=0, in which case position data is not VALID.

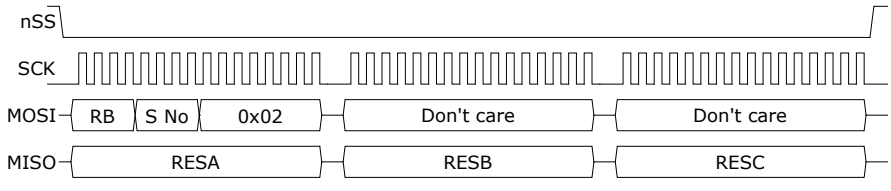


Figure 35 Write Read SPI transaction accessing RESA, RESB and RESC, Burst SPI Format

When using read only SPI transactions, the host can no longer clear Sample Indicators each SPI transaction. If Sample Indicators are still required, the host should configure them to a*uto clear before measurement*

(SAUTOCLR=1), which will result in a sample indicator being cleared before being activated by the next measurement, so that its behaviour follows that shown in Figure 34.

In pipeline measurement mode, the duration of the Echo is fixed. The remainder of the time it is performing the Pulse, during which the excitation coil is being driven continuously, drawing current from the excitation supply. When the Pipeline Interval is short the system is relatively efficient. However for longer Pipeline Intervals (greater than 700µs, say) the supply energy per pulse is significantly higher and the system becomes inefficient. This can be seen in Figure 10. In this case, it is recommended to use repeated single shot measurements instead.

9.8 Incremental Operation in Single Shot Mode

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. Fine coils deliver precise *fine position*, but are only incremental. The CTU combines the two position measurements (Figure 36), so that the position measurements that it reports to the host are always both precise and absolute.

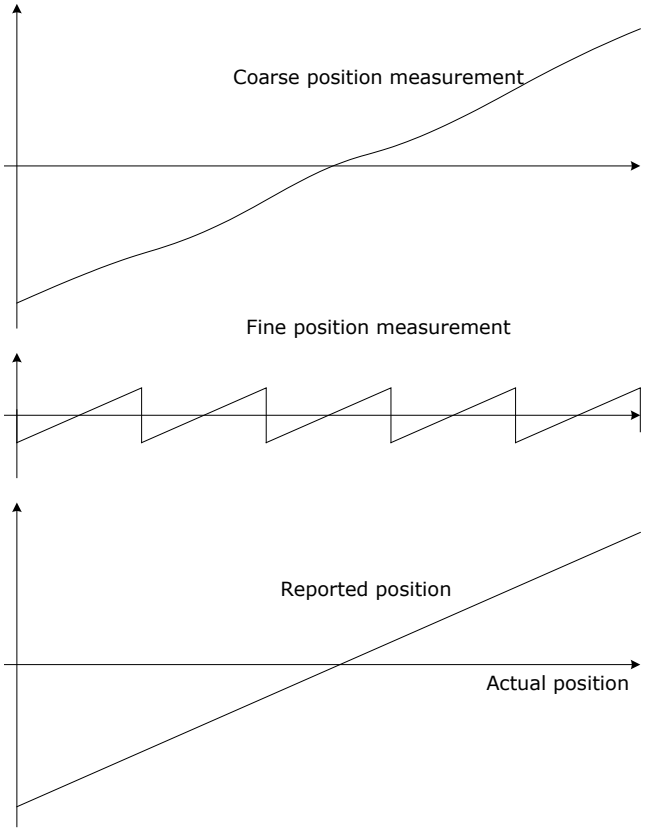


Figure 36 The CTU combines fine and coarse measurements

When measuring such sensors, the CTU has two modes of operation: *Absolute* and *Incremental*. In Absolute mode, reported position is based on adjacent fine and coarse measurements. In Incremental mode, the CTU uses fine position readings 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 sensor having SinLengthA=50mm, the limit is 25mm. When operating at a sample interval of 1ms, this means a speed of greater than 20m/s, 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.

9.9 Incremental Operation in Pipeline Mode

Pipeline measurement is optimised for fast update rates, and this is achieved by operating in Incremental mode for the large majority of measurements. An occasional Absolute mode measurement from the coarse coils may be performed, however, in order to recover from any fine period skipping error. There are two mechanisms provided for this. The host may request an Absolute measurement by setting INCE=0 for a single measurement. Alternatively, the host may set INCE=0 for all measurements, and use a sensor's ABSSEL register to specify ABSCOUNT, the number of Incremental mode measurements to take between coarse measurements. The relationship between ABSSEL and ABSCOUNT is defined in Equation 4 of section 8.11.

Figure 37 illustrates the case where the host requests an occasional Absolute measurement during pipeline measurement by writing INCE=0 using a write read SPI transaction, here labeled "B". The host must also use a write read SPI transaction for the following sample to set INCE=1 again, here labeled "C". The other SPI transactions may be read only. The CTU responds by performing an Absolute Measurement between SPI transactions C and D, and reporting the result at SPI transaction D. The INCF bit will be 0 to indicate that an Absolute measurement was taken. Reported Position will be the same as the previous transaction, unless a *fine period skipping error* occurred in which case Reported Position will be corrected accordingly.

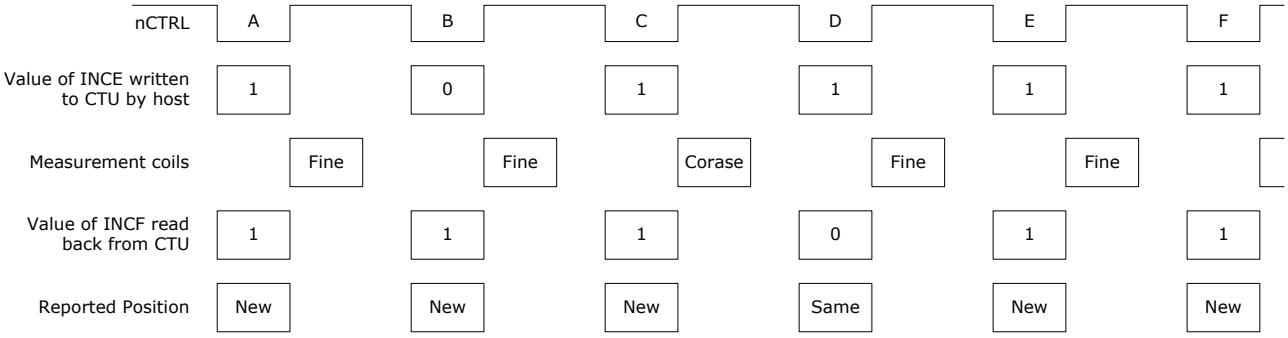


Figure 37 Host clears INCE once to perform Absolute measurement

Figure 38 illustrates the case where the CTU counts a number of Incremental measurements between each Absolute measurement. Before starting Pipeline measurement, the host should write the value of ABSSEL required to obtain the desired value of ABSCOUNT to the sensor's ABSSEL bits in its CONST register (section 8.11). The host should then start Pipeline measurement, but this time with INCE=0. The sensor's INCF bit indicates whether the SPI transaction's results were made by an Incremental measurement (INCF=1) or an Absolute measurement (INCF=0). In the example shown, the results returned in transactions B and Y are Absolute and the remainder are Incremental. The default value of ABSSEL is 12, representing 4096 samples or 800ms of operation running at 200µs Pipeline Interval. An ABSSEL value of 0 yields ABSCOUNT=1 and hence alternating Absolute and Incremental measurements.

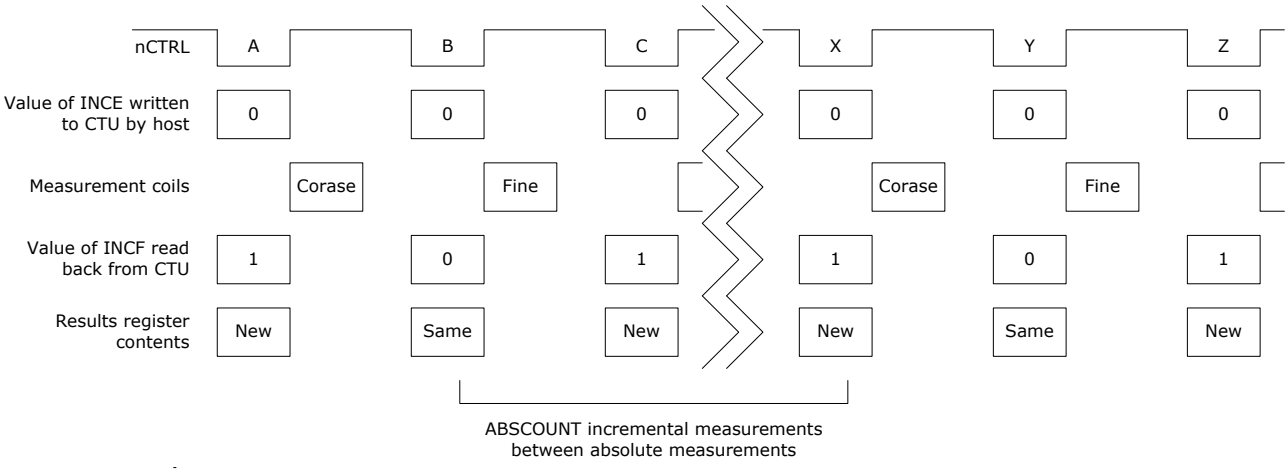


Figure 38 CTU counts number of Incremental measurements between Absolute

Since pipeline measurement allows a higher sample rate than single shot, a higher maximum velocity can be reliably detected. A fine period skipping error can be avoided providing the change in position between fine measurements is less than half the fine pitch between fine measurements. Note that an occasional coarse measurement doubles the maximum time between fine measurements in pipeline mode, so for operation at the highest possible velocity coarse measurements should be avoided (keep INCE=1). For example a Type 6.5 sensor may be operated at 24,000 rpm at

a Pipeline Interval of 200µs with INCE=1 and a 20% safety margin, and this drops to 10,000rpm for any measurement undertaken with INCE=0, allowing additional margin for coarse measurement error.

9.10 Continuous Measurement

In Continuous Measurement mode the CAM502 samples continuously without the host triggering each measurement with an SPI transaction. The Sample Interval is timed by the CAM502, and is host configurable with the SYSI register (section 8.2).

Continuous Measurement is initiated by writing CONT=GO=1 to a sensor's SCW register. Once initiated, the CAM502 samples continuously until stopped (CONT=GO=0).

For Sample Intervals less than 1ms the CAM502 should be operated in Continuous Pipeline Mode (CONT=GO=PIE=1). The CAM502 will now sample at high speed, using the same approach as described in section 9.7, except with measurements triggered internally instead of by the falling edge of the nCTRL/nSS lines.

If a change to the Sample Interval is required while operating in Continuous Pipeline Mode, the host should stop measurements (CONT=GO=PIE=0), then change the value of SYSI, then restart (CONT=GO=PIE=1).

During Continuous Measurement, the CAM502's results may be read over the SPI interface, at any time consistent with the SPI interface timing parameters of Table 15.

It is possible to omit reads from the SCW and PTEF registers for higher speed, as illustrated in Figure 35. In this case it is important to monitor the Amplitude A register (RESC for a Type 2 or Type 6 sensor). An Amplitude reading less than the minimum for VALID (Table 13) may be interpreted as VALID=0, in which case position data is not VALID.

When performing continuous measurements, it is recommended to configure the CAM502 to perform an occasional absolute mode measurement as described in 9.9, with INCE always 0 and ABSSEL set to the desired frequency of coarse measurements according to Equation 4 of section 8.11. This enables the CAM502 to recover from any fine period skipping error due to excess velocity, as described in sections 9.8 and 9.9.

The SPI timing parameter TAD must be extended when operating in Continuous Pipeline Mode and using Standard SPI Format. TAD is specified in Table 15. This is to allow the CAM502 more time to respond to the first (address) word of the SPI transaction, while also taking measurements asynchronously. If SPI timings then become excessive, Burst SPI Format may be used instead (section 7.4). This speeds up SPI transactions, since the register address is then specified in the previous SPI transaction. Please see section 7.6 for timing examples.

Continuous Pipeline Mode may be used in conjunction with Position Filtering to deliver reduced position noise and hence improved Noise Free Resolution, as described in section 9.11.

9.11 Position Filtering

By default, the position value reported by the CAM502 is always the result of the most recent measurement. This yields the shortest possible Group Delay between actual and reported position. In this case Noise Free Resolution is mainly a function of Amplitude, for example as shown in section 4.4.

The CAM502 may instead be configured to perform position filtering. This yields position data having lower noise content and therefore improved Noise Free Resolution, at the expense of an increase in Group Delay.

The CAM502's digital filter is simple single pole IIR which is based on the following equation:

$$Filtered\ Position\ (n) = Filtered\ Position\ (n-1) + k[Raw\ Position(n) - Filtered\ Position\ (n-1)]$$

Equation 5

k is the filter parameter which is host selectable with the FCSEL bits of a sensor's CONST register (section 8.11). Table 25 lists available settings, and their approximate effect on Noise Free Resolution and Group Delay. It also includes the value of Group Delay at a sample interval of 200µs in Pipeline Mode.

Table 25 Position Filtering Control

FCSEL	k	Noise Free Resolution Improvement / Bits	Increase in Group Delay /samples	Group Delay at 200us sample interval
0	1 (no filtering)	0	0	130µs
1	0.707	0.2	0.4	210µs
2	0.5	0.5	1	330µs
3	0.354	0.7	1.8	490µs
4	0.25	0.9	3	730µs
5	0.177	1.1	4.7	1.1ms
6	0.125	1.4	7	1.5ms
7	0.0884	1.6	10	2.1ms
8	0.0625	1.8	15	3.1ms
9	0.0442	2.1	22	4.6ms
10	0.0313	2.3	31	6.3ms
11	0.0221	2.5	42	8.5ms
12	0.0156	2.8	63	13ms
13	0.0110	3	84	17ms
14	0.00781	3.2	127	26ms
15	0.00552	3.4	255	51ms

Results are communicated over the SPI interface in the same way whether filtering is active or inactive, see section 8.10. Interface Resolution limits the achievable Noise Free Resolution, when Amplitude is high so that unfiltered noise is low and/or the value of FCSEL is high. It value is given by…

$$Interface\ Resolution = log_2\left(\frac{Measuring\ Length}{SinLength}\right) + 16\ bits$$

Equation 6

For a rotary Type 6 sensor where Measuring Length = 360° and Subtype = 360°/Subtype, this is the same as…

$$Interface\ Resolution = log_2(Subtype) + 16\ bits$$

Equation 7

9.12 Measurement Timing Specifications

Table 26 lists the values of timing parameters related to measurements, as defined in the preceding subsections. Timings are for when the CAM502 has detected an in-range target (VALID=1).

Different timing parameters are relevant depending on whether the CAM502 is performing pipeline measurements (PIF=1) for high-speed operation, or operating in single shot mode (PIE=0).

When a measurement includes reading from coarse coils, indicated by INCF=0, timings are generally longer.

When taking single shot measurements with position filtering active (FCSEL>0), the CAM502 takes longer to perform single shot measurements than without (FCSEL=0).

Table 26 measurement timing parameters

Parameter	Condition / Comment	Min	Max
Pipeline Interval (PIF=1)	Time between start of successive SPI transactions, CONT=0 Or requested sample interval (SYSI/10).µs, CONT=1	195µs	1600µs
Group Delay	INCF=1, FCSEL=0		140µs
TResults2nCTRL (PIF=1)	Timing margin between data available and low edge of nSS in the SPI transaction clocking the data out	2µs	
Measurement Time (PIE=0)	INCF=1, PIE=0, FCSEL=0		300µs
	INCF=1, PIE=0, FCSEL>0		310µs
	INCF=0, PIE=0		900µs
Sample Interval (PIE=0)	INCF=1, PIE=0	620µs	
	INCF=0, PIE=0	1240µs	
	CONT=1		13ms

9.13 Configurable Defaults

The default values of the CAM502 CTU chip's 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 CTU's registers. The host should then write 0x0C1C to the SAVEKEY register (section 8.5) followed by a 1 to the SAVE bit (section 8.1). The CTU will then reset itself. After this reset, and each subsequent one, the CTU's operation is determined by the saved Configurable Defaults.

The CTU's Factory Defaults can be restored by writing 0x0C1C to the SAVEKEY register followed by a 1 to the FACTORY bit (section 8.1). The CTU 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 CAM502 is manufactured.

Configurable Defaults revert to factory settings after an update of CTU Application Firmware (section 10). 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.

9.14 Status Codes

In normal operation the first word of each SPI transaction begins with the CTU's System ID, SYSID (section 8.8). Other codes may be encountered instead, and their meanings are described in Table 27.

Table 27 Status codes

First word of SPI transaction	Status description
SYSID (0xABCD by default)	Normal operation
0x10AD	CTU is in Bootloader mode expecting Application Code over the SPI interface
0x0BAD	CTU is in Bootloader mode and has detected SPI transactions that do not include valid Application Code
0x600D	Bootloader data block transfer has been successful
0xE001	External clock failed to start. Check crystal or external clock source present.
0xE002	External clock stopped. Check crystal or external clock source present.
Other words starting 0xE	Reserved

10 Bootloader Operation

10.1 Overview

The CTU chip's internal software is partitioned into two fields: Application Code and Bootloader Code. The Application Code is responsible for normal CTU operation including measurements, with communication through the register interface described in section 8. The Bootloader Code can be used to update the Application Code. 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 8.7). The version number of the Bootloader Code can be read from the BOOTVER register (section 8.6).

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 CTU chip over the SPI interface using CambridgeIC's CTU Adapter.

The CAM502BE-0003 chip does not include application code. Once parts have been mounted on a customer PCB at manufacture, they must be programmed with Application Code before normal operation commences.

10.2 Applications

Updating Application Code can be useful for adapting CTU chips to work with newly supported sensors, sensing circuits and peripherals. Instead of buying a new stock of chips, a customer can program existing chips with the new code.

10.3 CTU Firmware File Format

New CTU 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, 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 <CR><LF> (0x0D, 0x0A). The data block can be identified by searching for the first character after <CR><LF> that is not "#". This is the first character of the data block.

The Data Block comprises the remaining characters in the .cff file, except for the last two, which are the Checksum Word for the Data Block. The Checksum Word may be used to verify that the Data Block is valid and uncorrupted. It is strongly recommended that the host system checks that the checksum is valid before updating Application Code.

10.4 Choice of Bootloader Method

There are two Bootloader methods. The simplest and recommended approach requires the host to control the CTU's nRESET line in real time, and is described in section 10.5. An alternative is provided when the host does not have control over the nRESET line, described in section 10.6.

10.5 Bootloader Operation With nRESET

This method is illustrated in Figure 40. To start, the host toggles nRESET to reset the CTU. It then begins the Data Block Transfer Process described in section 10.7. For correct operation, a minimum time period is required between the reset and first SPI transaction, TnRST2nSS, illustrated in Figure 39 and specified in Table 28. The second word of the Data Block must have been sent within a time TVALIDCHECK(Min) of the reset, otherwise the CTU may resume normal operation instead of updating Application Code. TVALIDCHECK is specified in Table 15.

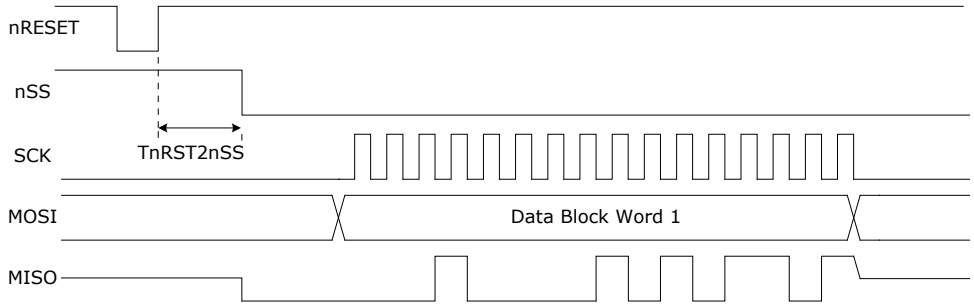


Figure 39 Timing of first Data Block Word SPI transaction after

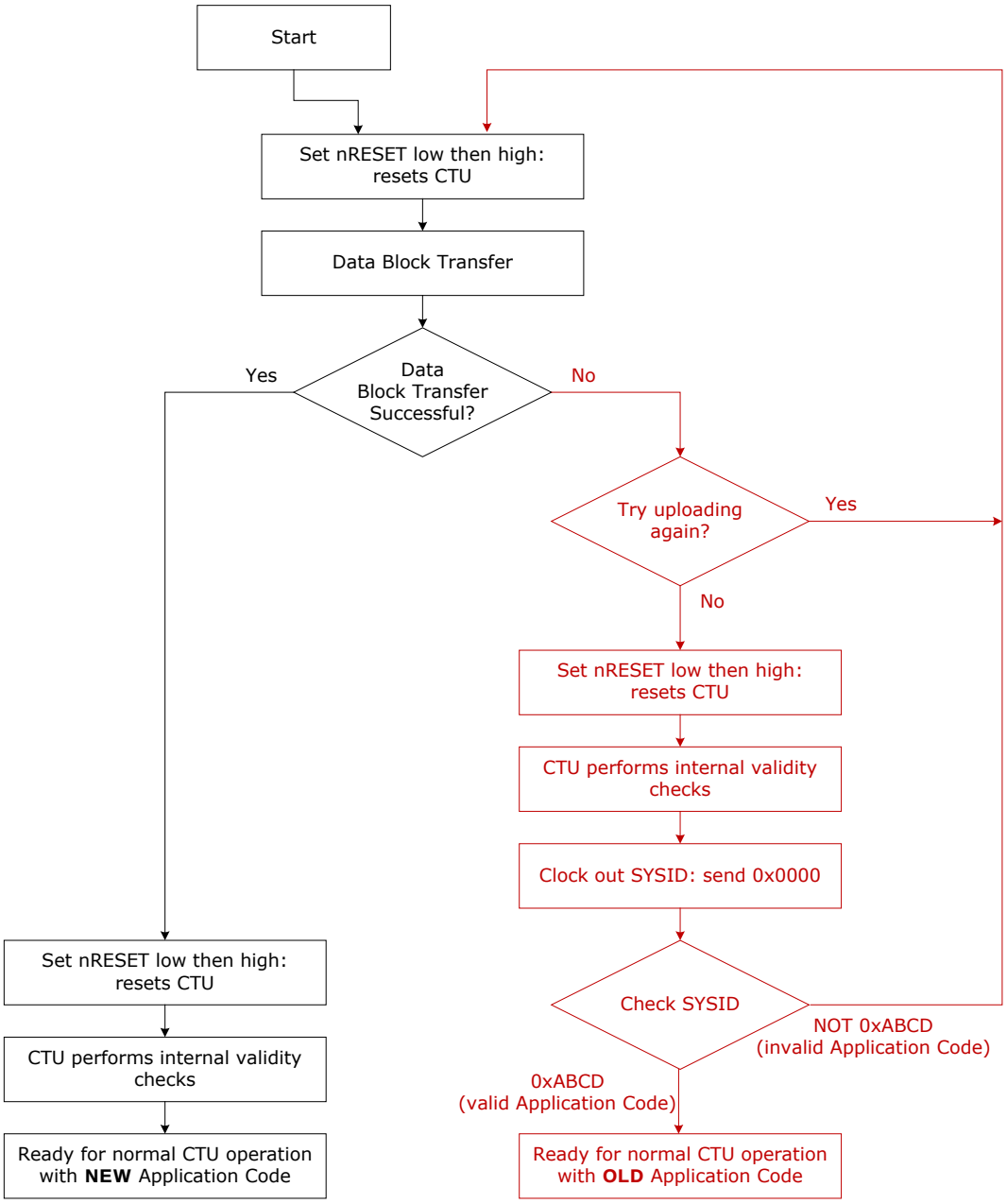


Figure 40 Bootloader Method, with nRESET

If the Data Block Transfer is successful, the host should reset the CTU, wait for validity checks to complete (section 9.1) and resume normal operation. It is recommended that the host checks that the System Version Number reported in the SYSVER register matches the expected value for the new Application Code.

If the Data Block is not successful, the host can repeat the process. If the upload failed early in the process, it is possible that old Application Code will remain valid. The host can attempt to run old Application Code, by following the process illustrated in red shown in Figure 40. Valid Application Code is indicated by SYSID equal to 0xABCD (the factory default) or by an alternative valid value programmed as a Configurable Default.

10.6 Bootloader Operation Without nRESET

This method is illustrated in Figure 42. Before Data Block Transfer can begin, the CTU must be put into its Bootloader Mode. There are two methods, depending on whether or not the CTU contains valid Application Code already.

If the CTU contains valid Application Code, the host can set the BOOTLOAD bit in the SYSCW register while the CTU is operating normally (section 8.1). The CTU will reset itself, and when it comes out of reset it will remain in Bootloader Mode. The host can then perform the Data Block Transfer process described in section 10.7 to load new Application Code. The first Data Block Word should be timed as in Figure 41, so that it does not violate the minimum time TnRST2nSS specified in Table 28.

If the Data Block Transfer is successful, the host should then send the Boot Command: 0xB002, to the CTU. The CTU will reset itself and begin validity checks on the new Application Code. The host should wait for these checks to complete before resuming normal operation (see section 9.1). It is recommended that the host checks that the System Version Number reported in the SYSVER register matches the expected value for the new Application Code.

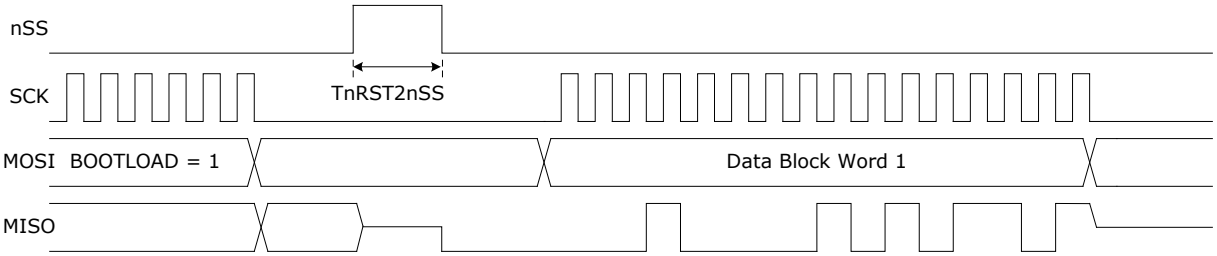


Figure 41 Timing of first Data Block Word SPI transaction after BOOTLOAD bit set

If the Data Block Transfer process fails, the host should send the Boot Command and wait for the CTU's internal validity checks to complete, as illustrated in Figure 42. If the CTU reports the default SYSID (factory default 0xABCD or as set up as a Configurable Default) the old Application Code remains valid. In this case, the host can attempt the upload process again using the BOOTLOAD bit to enter Bootloader Mode as above.

If the Data Block Transfer process fails and the SYSID test does not result in the expected value, then the CTU's old Application Code has become corrupted. The BOOTLOAD bit can not be used to enter Bootloader Mode since the register map is defined in Application Code. Without host control over nRESET, the only remaining way to enter Bootloader Mode is to power cycle the CTU. The host should then begin Data Block Transfer, subject to the minimum timing TnRST2nSS specified in Table 49. There is no upper limit on the time between power cycling and the start of Data Block Transfer. However, the host must not send any data to the CTU over its SPI interface before Data Block Transfer, otherwise it will be interpreted as the Data Block and a subsequent upload will fail.

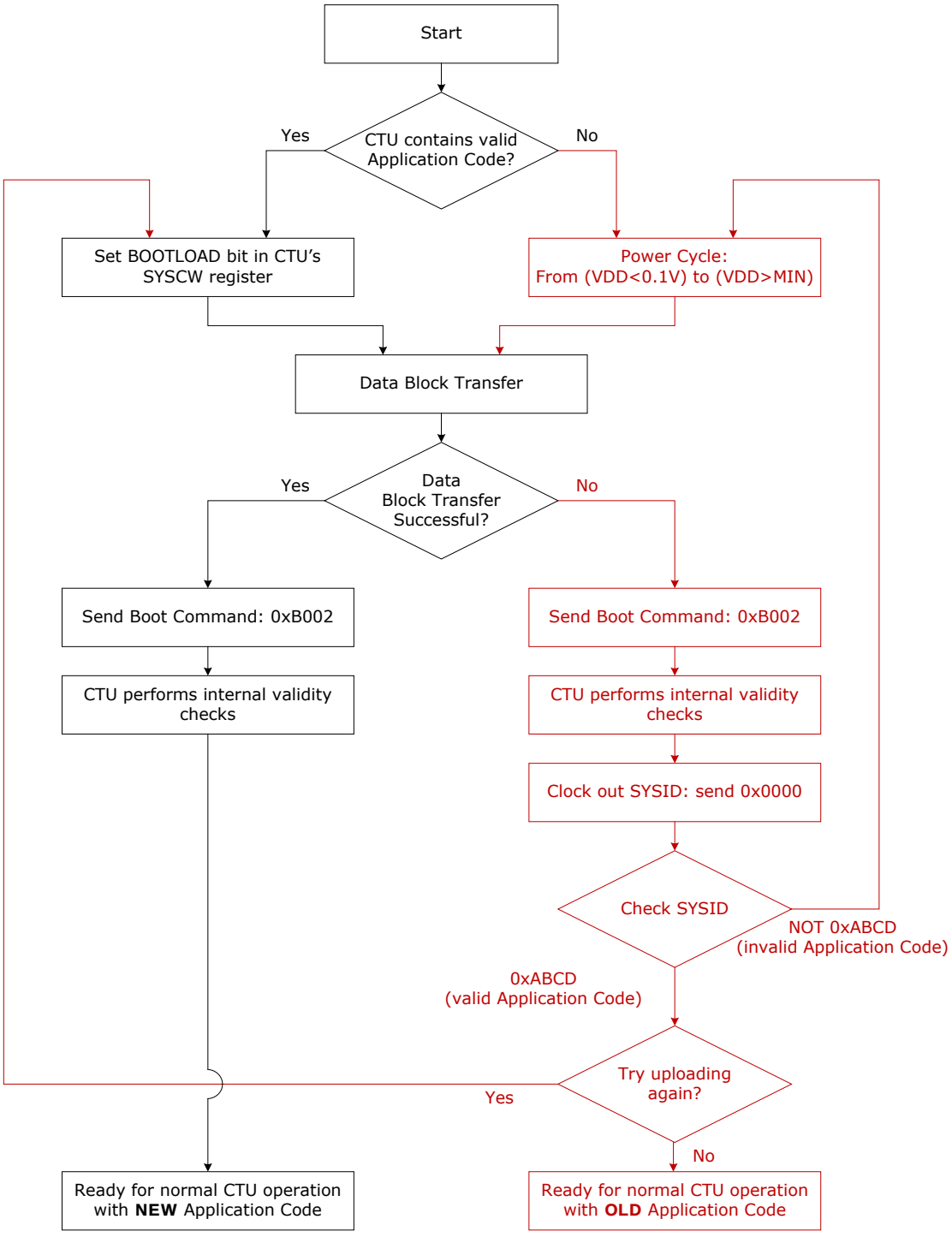


Figure 42 Bootloader method, without nRESET

10.7 Data Block Transfer Process

The Data Block Transfer process is for sending new Application Code to the CTU, and is illustrated in Figure 43. It forms part of both Bootloader Methods described in section 10.5 and section 10.6.

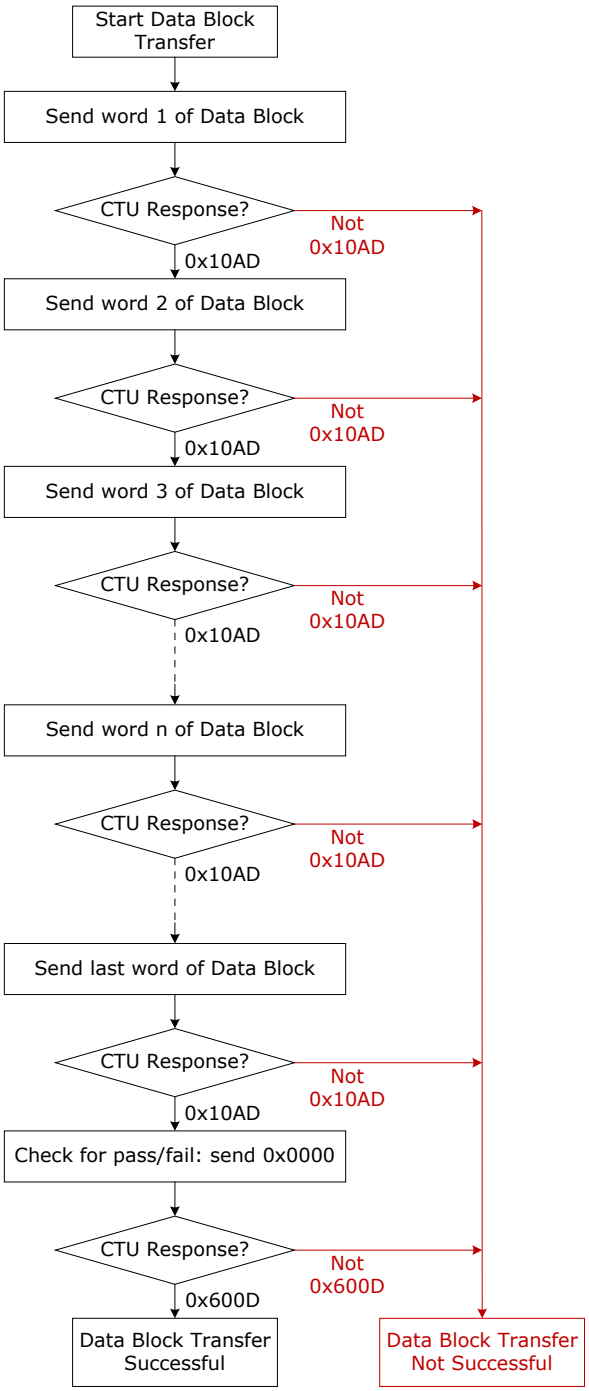


Figure 43 Data Block Transfer Process

Data is transferred in 16-bit words over the SPI interface, as described in section 10.8. Each time the host sends a word, the CTU should respond with 0x10AD. If not, Data Block Transfer has failed. When the last word of the Data Block has been sent and the CTU has signaled it is ready as described in section 10.8, the host can check whether the Data Block Transfer was successful. It does this by sending 0x0000, or by sending the data block's checksum, and checking the response from the CTU is 0x600D. If not (typically responding with 0x0BAD) the Data Block Transfer has failed.

10.8 SPI Communication with the CTU in Bootloader Mode

Communication with the CTU in Bootloader Mode uses the same SPI timings and bit ordering as described in section 7.

There is one difference: the MISO output from the CTU is used for handshaking. After each Data Block word, the CTU sets MISO to high impedance (it will normally be pulled high by a pull-up resistor). When the CTU is ready for the next Data Block word, it pulls MISO low. The host should detect the state of MISO and only send the next Data Block word when it has been pulled low by the CTU.

This behaviour is illustrated in Figure 44. The time between the last clock edge of a Data Block word and the CTU signaling it is ready for the next Data Block word is denoted TBOOTWAIT. The maximum value of TBOOTWAIT is specified in Table 28. The maximum value shown is only required on a small number of Data Block words when the CTU is performing internal tests. In most cases it is much smaller.

The Data Block Transfer process may be undertaken as a single SPI transaction with nSS low throughout. Alternatively, Data Block words or groups of words may be sent as individual SPI transactions with nSS high in between. In all cases the host should check MISO as described above before sending the next Data Block word.

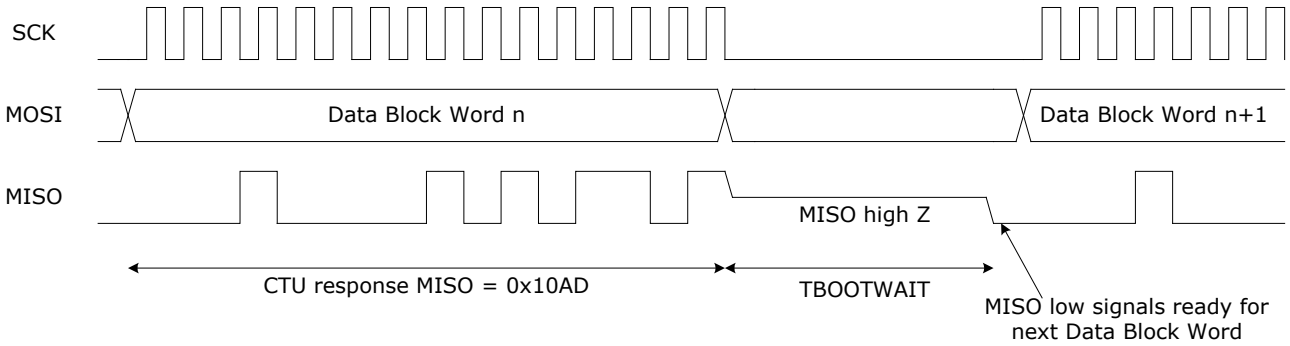


Figure 44 SPI communication, sending Data Block

10.9 Bootloader Timing Specifications

The specifications for the Bootloader timings referred to above are in Table 28.

Table 28 Bootloader timings

Parameter	Description	Min	Typ	Max	Units
TBOOTLOAD	Overall time to update Application Code	-	2.6	-	s
TBOOTWAIT	Variable pause required between Data Block Words	-	0.009	300	ms
TnRST2nSS	Pause required before start of Data Block Transfer	20	-	-	µs

11 Package Details

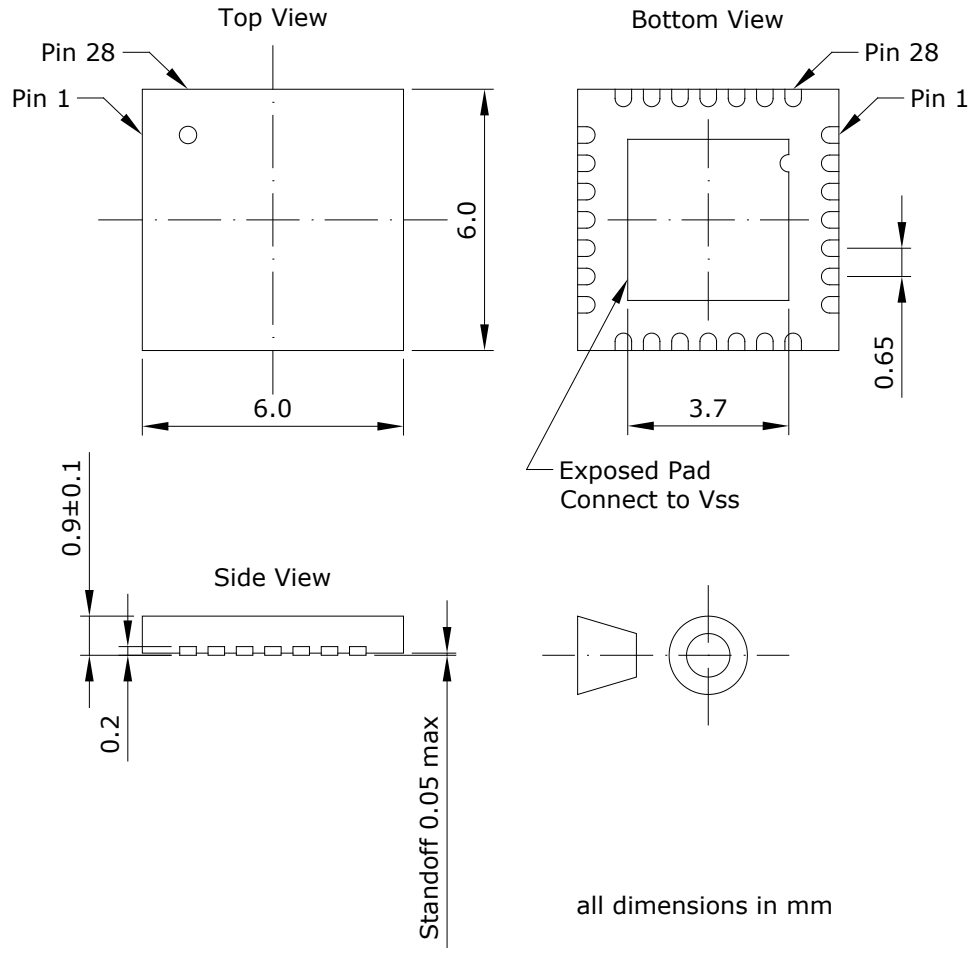


Figure 45 28-pin QFN package

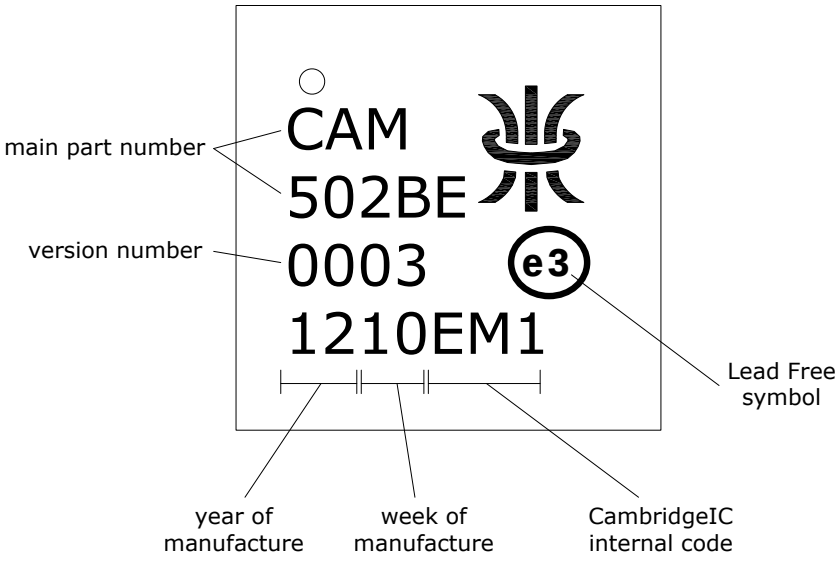


Figure 46 product markings

12 Tape and Reel Specifications

CAM502 chips are available in tape and reel on complete reels of 1600 parts. The carrier tape is illustrated in Figure 47, and dimensions are specified in Table 29.

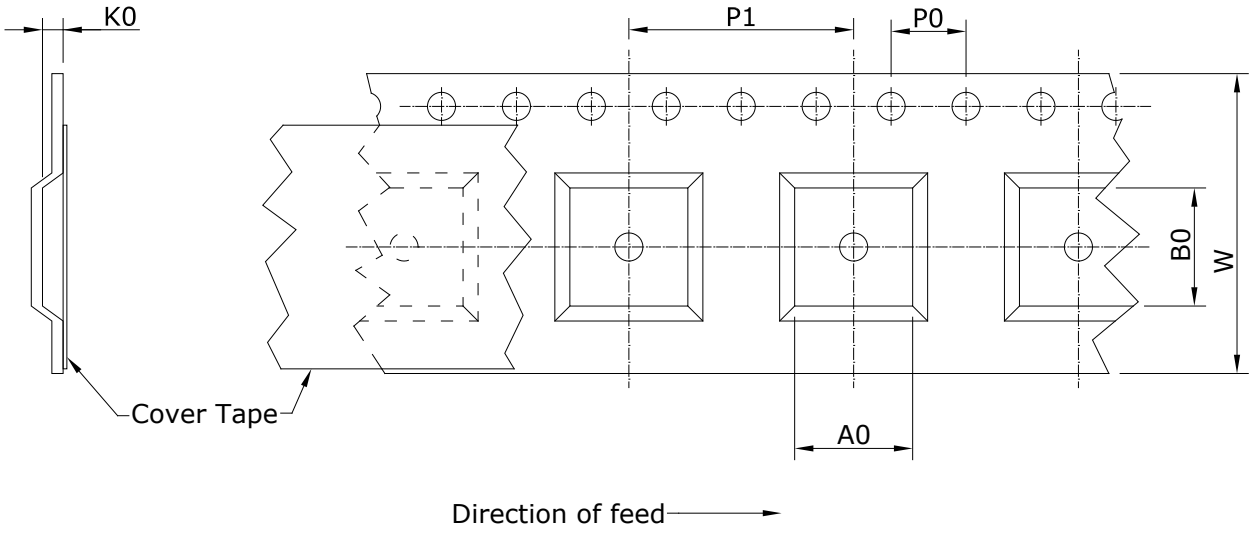


Figure 47 Carrier tape dimensions

Table 29

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

13 Reflow Soldering Recommendations

The CAM502 is available in lead free packaging only. The recommended reflow soldering temperature profile is illustrated in Figure 48. Values are shown in Table 30.

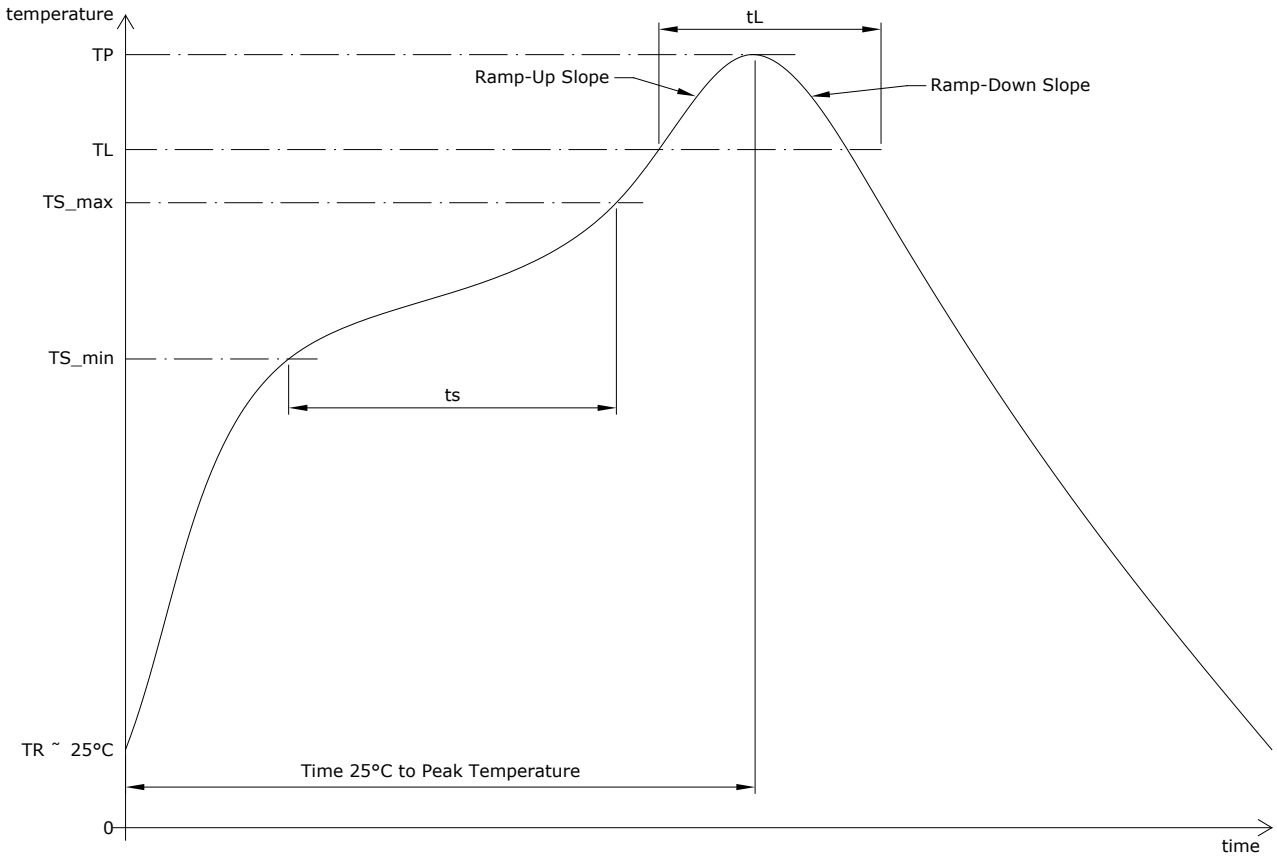


Figure 48 reflow soldering temperature profile definitions

Table 30 reflow soldering parameters

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		

14 Environmental

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

15 RoHS Compliance

The CAM502 uses Matte Tim (Sn) pin finish. CambridgeIC certifies, to the best of its knowledge and understanding, that the CAM502 chip is in compliance with EU RoHS, China RoHS and Korea RoHS.

16 Document History

Revision	Date	Description
0008	26 November 2012	Applies to SYSVER ≥ 0x0112 Added details of operation with Type 6 sensors Added circuit layout recommendations Expanded System Operation and Measurement Timing section Added Electrical Characteristics Added Status Codes
0009	18 December 2014	Applies to SYSVER ≥ 0x0116 Reduced resonator VALID frequency range by 0.2kHz each end. Split ABSCOUNT register into new fields and renamed CONST. ABSSEL now selects ABSCOUNT value. New FCSEL field in CONST allows host to request filtered position sample, and determines how much filtering to apply. Modified measurement timing parameters, including those resulting from FCSEL. Changed definition of SYSI to represent multiples of 0.1µs instead of 1µs when PIE=1, and of 100µs when PIE=0. Added Burst SPI Format description and SPI register access examples. Added SPI timing parameters related to new features above.

17 Contact Information

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

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

18 Legal