IQS7320A DATASHEET

4-Channel Inductive IC for Keyboard/Keypad applications

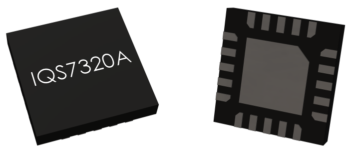
1 Device Overview

The IQS7320A ProxFusion® IC is a multi-channel inductive sensing device that is mainly aimed at applications that require multi-level trigger points and fast report rates. The trigger-level UI allows for adjustable trigger point sensitivity resulting in a better user experience with best-in-class sensitivity and power consumption. Other features include automatic tuning and long-term environmental tracking.

1.1 Main Features

- Highly flexible ProxFusion® device
- Up to 4 inductive sensors
- Adjustable trigger point detection for each sensor
- Greater than 1kHz report rate
- Synchronous matrix column key scanning
- RF Immunity
- Sensor flexibility
  - Automatic sensor tuning for optimum sensitivity
  - Internal voltage regulator
  - On-chip noise filtering
  - Trigger point detection hysteresis
- Configurable power modes for optimal response rate vs power consumption
- I^{2}C communication interface with IRQ/RDY (up to fast plus - 1MHz)
- QFN20 (3 x 3 x 0.5 mm) – 0.4mm pitch
- Wide input voltage supply range: 2.2V to 3.5V
- Wide operating temperature range: -40°C to +85°C

RoHS2
Compliant



QFN20 package
Representation only

1.2 Applications

- Mechanical Keyboards
- Trackpad Force Touch
- Gaming Controllers
- Waterproof Buttons (Inductive)
- Remote Controls

1.3 Block Diagram

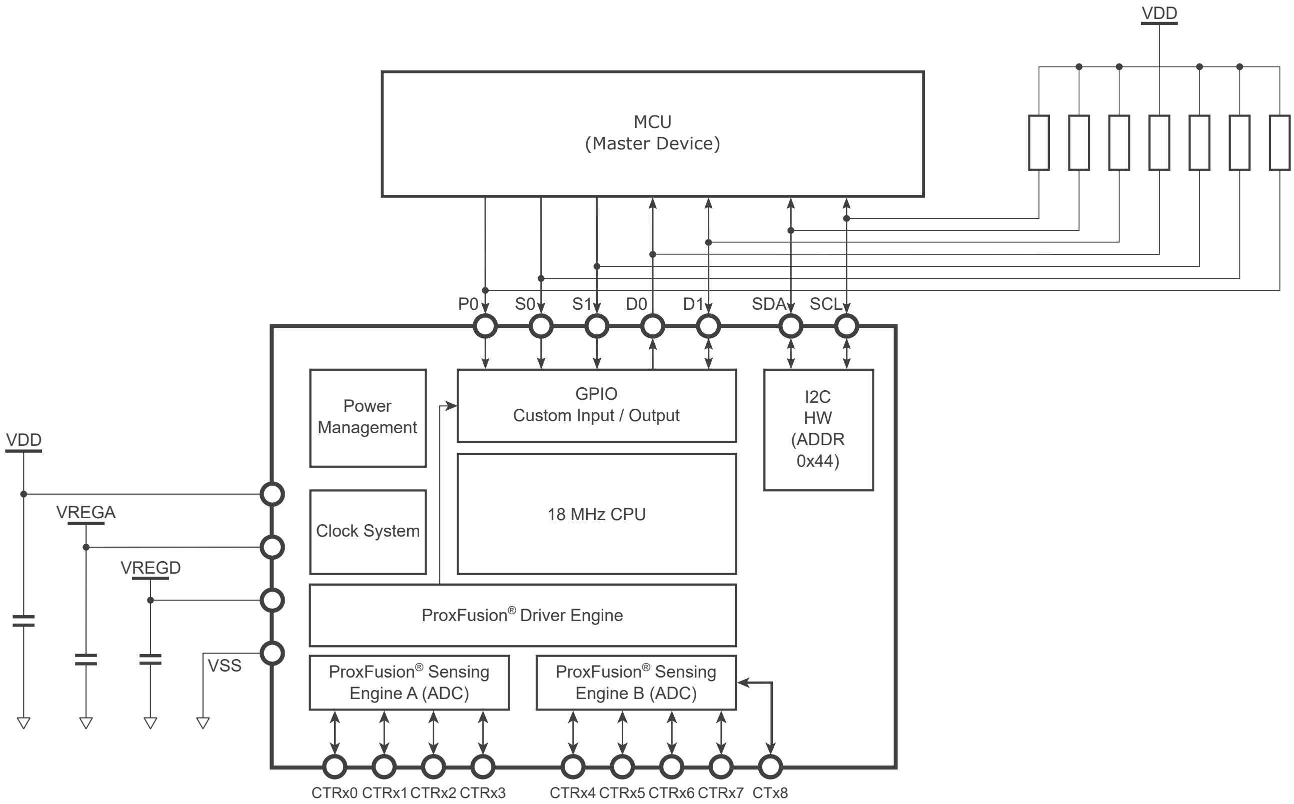


Figure 1.1: Functional Block Diagram[i]

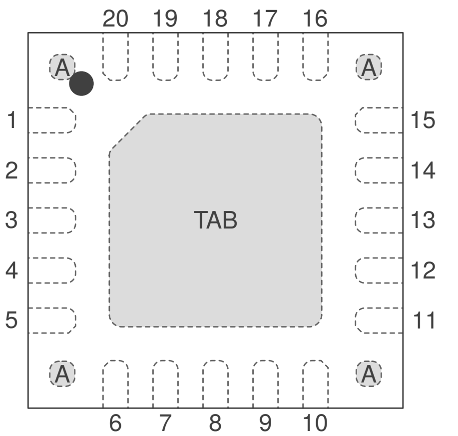
[i] WLCSP18 packages do not have a CRx4 and combines GPIO0 and GPIO3

Contents

2 Hardware Connection

2.1 QFN20 Pin Diagram

Table 2.1: 20-pin QFN Package (Top View)



Top View

Pin no.	Signal name	Pin no.	Signal name
1	VDD	11	CRx6/CTx6
2	VREGD	12	CRx7/CTx7
3	VSS	13	CTx8/Vbias
4	VREGA	14	S0/GPIO0
5	CRx0/CTx0	15	S1/GPIO3
6	CRx1/CTx1	16	D0/GPIO4
7	CRx2/CTx2	17	D1/GPIO5
8	CRx3/CTx3	18	SCL/GPIO2
9	CRx4/CTx4	19	SDA/GPIO1
10	CRx5/CTx5	20	MCLR/GPIO6

Area name	Signal name
TAB[i]	Thermal pad (floating)
A[ii]	Thermal pad (floating)

2.2 Pin Attributes

Table 2.2: Pin Attributes

Pin no. QFN20	Signal name	Signal type	Buffer type	Power source
1	VDD	Power	Power	N/A
2	VREGD	Power	Power	N/A
3	VSS	Power	Power	N/A
4	VREGA	Power	Power	N/A
5	CRx0/CTx0	Analog		VREGA
6	CRx1/CTx1	Analog		VREGA
7	CRx2/CTx2	Analog		VREGA
8	CRx3/CTx3	Analog		VREGA
9	CRx4/CTx4	Analog		VREGA
10	CRx5/CTx5	Analog		VREGA
11	CRx6/CTx6	Analog		VREGA
12	CRx7/CTx7	Analog		VREGA
13	CTx8/Vbias	Analog		VREGA
14	S0/GPIO0	Digital		VDD
19	SDA/GPIO1	Digital		VDD
18	SCL/GPIO2	Digital		VDD
15	S1/GPIO3	Digital		VDD
16	D0/GPIO4	Digital		VDD
17	D1/GPIO5	Digital		VDD
20	MCLR/GPIO6	Digital		VDD

[i] It is recommended to connect the thermal pad (TAB) to VSS.
[ii] Electrically connected to TAB. These exposed pads are only present on *–QNR* order codes.

2.3 Signal Descriptions

Table 2.3: Signal Descriptions

Function	Signal name	Pin no. QFN20	Pin type[iii]	Description
ProxFusion®	CRx0/CTx0	5	IO	ProxFusion® channel
	CRx1/CTx1	6	IO	
	CRx2/CTx2	7	IO	
	CRx3/CTx3	8	IO	
	CRx4/CTx4	9	IO	
	CRx5/CTx5	10	IO	
	CRx6/CTx6	11	IO	
	CRx7/CTx7	12	IO	
	CTx8/Vbias	13	O	CTx8 pad
GPIO	S0/GPIO0	14	IO	S0 - Configuration input select 0
	S1/GPIO3	15	IO	S1 - Configuration input select 1
	D0/GPIO4	16	IO	D0 - Channel state output 0 / Configuration input select 2
	D1/GPIO5	17	O	D1 - Channel state output 1 / Configuration input select 3
	MCLR/GPIO6	20	IO	Active pull-up, 200k resistor to VDD. Analogue conversions synchronization line. VPP input for OTP.
I^2C	SDA/GPIO1	19	IO	I^2C Data
	SCL/GPIO2	18	IO	I^2C Clock
Power	VDD	1	P	Power supply input voltage
	VREGD	2	P	Internal regulated supply output for digital domain
	VSS	3	P	Analog/Digital Ground
	VREGA	4	P	Internal regulated supply output for analog domain

[iii] Pin Types: I = Input, O = Output, IO = Input or Output, P = Power.

2.4 Reference Schematic

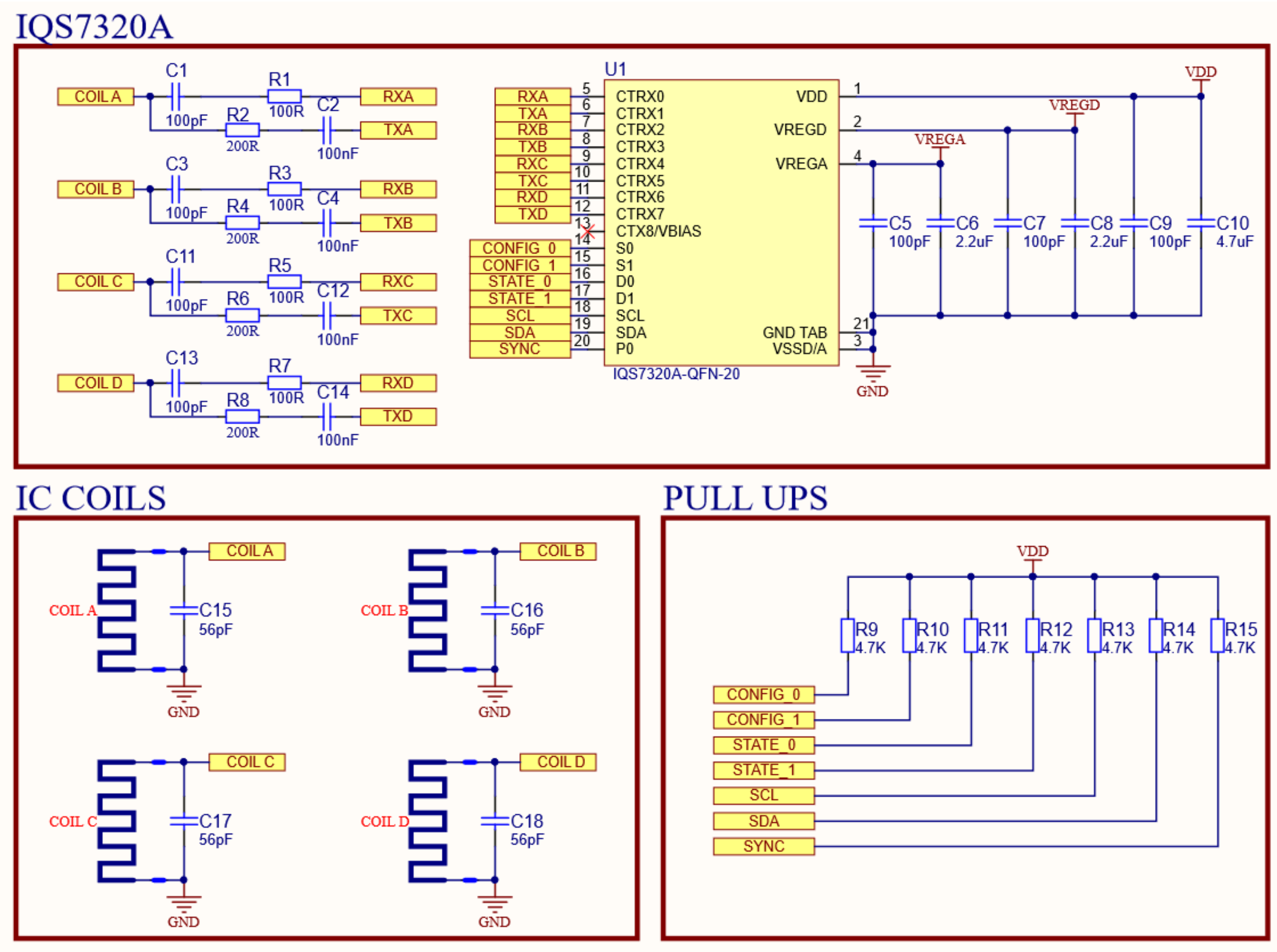


Figure 2.1: 4 Channel Inductive Sensor Reference Schematic

3 Electrical Characteristics

3.1 Absolute Maximum Ratings

Table 3.1: Absolute Maximum Ratings

	Min	Max	Unit
Voltage applied at VDD pin to VSS	2.2	3.5	V
Voltage applied to any ProxFusion® pin	−0.3	VREGA	V
Voltage applied to any other pin (referenced to VSS)	−0.3	VDD + 0.3 (3.5 V max)	V
Storage temperature, T_{stg}	−40	85	°C

3.2 Recommended Operating Conditions

Table 3.2: Recommended Operating Conditions

		Min	Nom	Max	Unit
VDD	Supply voltage applied at VDD pin: F_{OSC} = 18 MHz	2.2		3.5	V
VREGA	Internal regulated supply output for analog domain: F_{OSC} = 18 MHz	1.7	1.75	1.79	V
VREGD	Internal regulated supply output for digital domain: F_{OSC} = 18 MHz	1.75	1.8	1.85	V
VSS	Supply voltage applied at VSS pin		0		V
T_A	Operating free-air temperature	−40	25	85	°C
C_{VDD}	Recommended capacitor at VDD	2×C_{VREGA}	3×C_{VREGA}		µF
C_{VREGA}	Recommended external buffer capacitor at VREGA, ESR$\leq$ 200 mΩ	2	4.7	10	µF
C_{VREGD}	Recommended external buffer capacitor at VREGD, ESR$\leq$ 200 mΩ	2	4.7	10	µF
$Cx_{SELF-VSS}$	Maximum capacitance between ground and all external electrodes on all ProxFusion® blocks (self-capacitance mode)	1	-	400[i]	pF
$Cm_{CTx-CRx}$	Capacitance between Receiving and Transmitting electrodes on all ProxFusion® blocks (mutual-capacitance mode)	0.2	-	9[i]	pF
$Cp_{CRx-VSS-1M}$	Maximum capacitance between ground and all external electrodes on all ProxFusion® blocks (mutual-capacitance mode @ f_{xfer} = 1 MHz)			100[i]	pF
$Cp_{CRx-VSS-4M}$	Maximum capacitance between ground and all external electrodes on all ProxFusion® blocks (mutual-capacitance mode @ f_{xfer} = 4 MHz sensing)			25[i]	pF
$\frac{Cp_{CRx-VSS}}{Cm_{CTx-CRx}}$	Capacitance ratio for optimal SNR in mutual-capacitance mode[ii]	10		20	n/a
$RCx_{CRx/CTx}$	Series (in-line) resistance of all mutual-capacitance pins (Tx & Rx pins) in mutual-capacitance mode	0[iii]	0.47	10[iv]	kΩ
RCx_{SELF}	Series (in-line) resistance of all self-capacitance pins in self-capacitance mode	0[iii]	0.47	10[iv]	kΩ

3.3 ESD Rating

Table 3.3: ESD Rating

		Value	Unit
$V_{(ESD)}$ Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001[v]	±4000	V

[i] RCx = 0 Ω.
[ii] Please note that the the maximum values for Cp and Cm are subject to this ratio.
[iii] Nominal series resistance of 470 Ω is recommended to prevent received and emitted EMI effects. Typical resistance also adds additional ESD protection.
[iv] Series resistance limit is a function of f_{xfer} and the circuit time constant, RC. $R_{max} \times C_{max} = \frac{1}{(6 \times f_{xfer})}$ where C is the pin capacitance to VSS.
[v] JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process. Pins listed as ±4000 V may actually have higher performance.

3.4 Current Consumption

Mutual Inductive Setup: ATI Target = 320, F_{OSC} = 18MHz

Table 3.4: Current consumption

Communication Method	Power mode	Sample Period [ms]	Sync / Keyscan Period [ms]	Typical Current [µA]
Key Scanning	Normal	0.5	N/A	2650
Key Scanning	Low	1	N/A	2650
Key Scanning	Low	2	N/A	2430
Key Scanning	Low	5	N/A	910
Key Scanning	Low	10	N/A	545
Key Scanning	Low	20	N/A	470
Key Scanning	Low	50	N/A	430
Key Scanning	Low	100	N/A	415
Key Scanning with Sync on Key Scan	Normal	0.5	1	2430
Key Scanning with Sync on Key Scan	Normal	0.5	5	2080
Key Scanning with Sync on Key Scan	Normal	0.5	10	2040
Key Scanning with Sync on Key Scan	Normal	0.5	20	2015
Key Scanning with Sync on Key Scan	Low	5	10	650
Key Scanning with Sync on Key Scan	Low	5	20	585
Key Scanning with Sync on Key Scan	Low	5	50	550
Key Scanning with Sync on Key Scan	Low	10	20	540
Key Scanning with Sync on Key Scan	Low	10	50	500
Key Scanning with Sync on Key Scan	Low	10	100	485
I^2C Streaming	N/A	5	N/A	1270
I^2C Streaming	N/A	10	N/A	730
I^2C Streaming	N/A	20	N/A	365
I^2C Streaming	N/A	50	N/A	165
I^2C Streaming	N/A	100	N/A	100
I^2C Streaming	N/A	500	N/A	45
I^2C Streaming with Sync	N/A	N/A	10	915
I^2C Streaming with Sync	N/A	N/A	30	640
I^2C Streaming with Sync	N/A	N/A	60	570

4 Timing and Switching Characteristics

4.1 Reset Levels

Table 4.1: Reset Levels

Parameter		Min	Typ	Max	Unit
V_{VDD}	Power-up/down level (Reset trigger) – slope > 100 V/s	1.040	1.353	1.568	V
V_{VREGD}	Power-up/down level (Reset trigger) – slope > 100 V/s	0.945	1.122	1.304	V

4.2 Miscellaneous Timings

Table 4.2: Miscellaneous Timings

Parameter		Min	Typ	Max	Unit
f_{OSC}	Master CLK frequency tolerance 18 MHz	17.1	18	19.54	MHz
f_{xfer}	Charge transfer frequency (derived from f_{OSC})	42	500 – 1500	4500	kHz

4.3 Digital I/O Characteristics

Table 4.3: Digital I/O Characteristics

Parameter		Test Conditions	Min	Typ	Max	Unit
V_{OL}	SDA & SCL Output low voltage	I_{sink} = 20 mA			0.3	V
V_{OL}	GPIO[i] Output low voltage	I_{sink} = 10 mA			0.15	V
V_{OH}	Output high voltage	I_{source} = 20 mA	VDD – 0.2			V
V_{IL}	Input low voltage				VDD × 0.3	V
V_{IH}	Input high voltage		VDD × 0.7			V
C_{b_max}	SDA & SCL maximum bus capacitance				550	pF

[i] Refers to S0, S1, D0, and D1 pins.

4.4 I²C Characteristics

Table 4.4: I²C Characteristics

Parameter		VDD	Min	Typ	Max	Unit
f_{SCL}	SCL clock frequency	2.2 V, 3.3 V			1000	kHz
$t_{HD,STA}$	Hold time (repeated) START	2.2 V, 3.3 V	0.26			µs
$t_{SU,STA}$	Setup time for a repeated START	2.2 V, 3.3 V	0.26			µs
$t_{HD,DAT}$	Data hold time	2.2 V, 3.3 V	0			ns
$t_{SU,DAT}$	Data setup time	2.2 V, 3.3 V	50			ns
$t_{SU,STO}$	Setup time for STOP	2.2 V, 3.3 V	0.26			µs
t_{SP}	Pulse duration of spikes suppressed by input filter	2.2 V, 3.3 V	0		50	ns

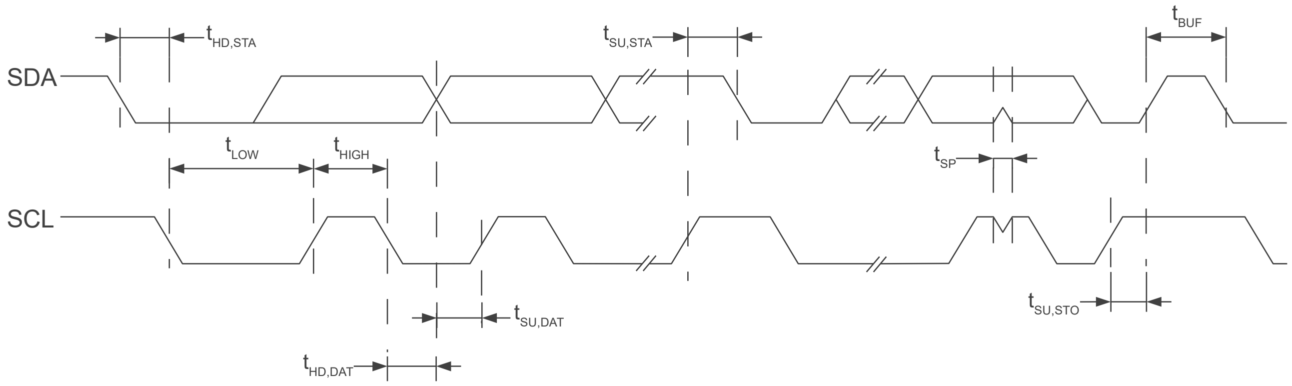


Figure 4.1: I²C Mode Timing Diagram

5 ProxFusion® Module

The IQS7320A contains dual ProxFusion® modules that use patented technology to measure and process the sensor data. Two modules ensure a rapid response from multi-channel implementations. The multiple filter-halt and trigger outputs are the primary output from the sensor.

5.1 Channel Options

The IC offers the following channel options to allow up to a maximum of 4 inductive channels:

- Two Inductive channels connected to the internal Prox engine A
- Two Inductive channels connected to the internal Prox engine B
- Two scanning cycles are available for measurements, with the IC able to sample on both Prox engine A & B simultaneously, which allows up to four inductive sensors to be sampled periodically.

- Inductive design layout guide: AZD115

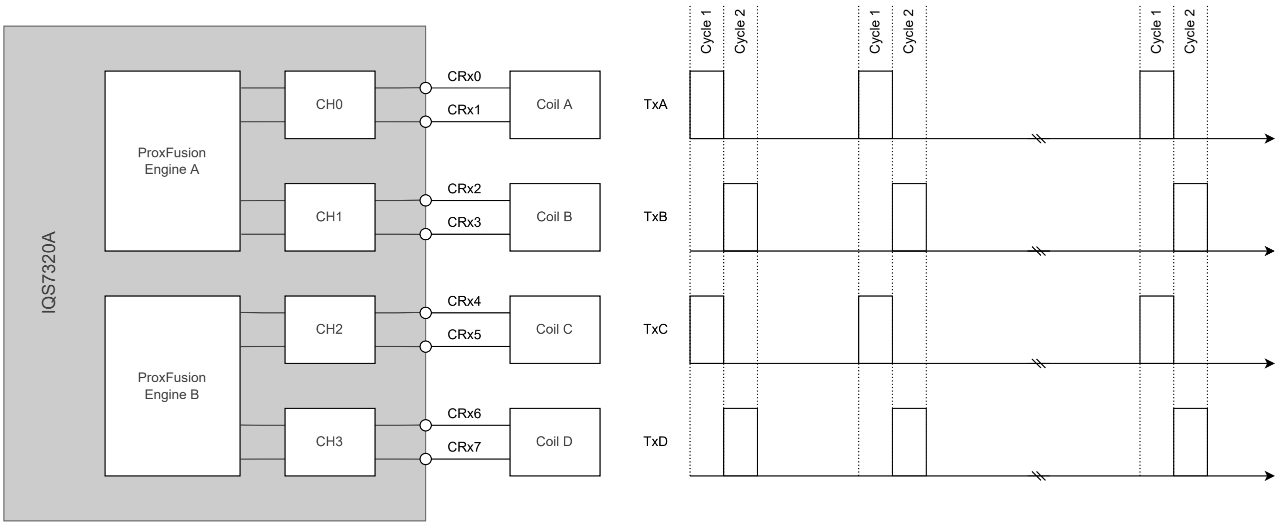


Figure 5.1: Dual ProxFusion Sensing

5.2 Count Value

The sensing measurement returns a count value for each channel. Count values are inversely proportional to the sensor inductance, and all outputs are derived from this value.

5.2.1 Max Count

Each channel is limited to having a non-linearized count value smaller than the 2047 count limit. If the measured counts exceed this limit, the conversion will be stopped, and the counts value will be equal to the limit.

5.2.2 Count Filter

The IQS7320A is aimed at high response rate applications to minimize input lag. To this end, the user only has the option to select that the filtered counts be an average of the last 1-4 raw counts values. This allows for some noise filtering with a high response rate. The number of averaged raw counts used to calculate the filtered counts can be set in the **global channel settings** register.

5.2.3 RF Block

The RF-blocking functionality of the device will ignore sampled counts for a limited number of cycles after the difference between two trailing samples exceed the **RF blocking threshold**.

5.3 Reference Value/Long-Term Average (LTA)

User interaction is detected by comparing the measured count values to some reference value. The reference value/LTA of a sensor is slowly updated to track changes in the environment and is not updated during user interaction.

5.3.1 Reseed

Since the reference for a channel is critical for the device to operate correctly, there could be known events or situations which would call for a manual reseed. A reseed takes the latest measured counts, and seeds the reference/LTA with this value, therefore updating the value to the latest environment. A reseed command can be given by setting the corresponding bit in the **system control** register.

5.3.2 LTA Filter

The IIR filter is applied to the sampled counts of a channel to calculate the LTA and offers various damping options as defined in the **filter beta settings**.

$$\text{Damping factor} = \frac{\text{Beta}}{256}$$

5.3.3 Filter-Halt

The device will attempt to stop updating the reference channels when user input is detected and will stop updating the channel reference when the difference between the sampled counts and the LTA exceeds the **filter-halt threshold**. This will result in the relevant channel's filter-halt event bit being set in the **events** register. The **filter-halt timeout** value can be adjusted to set the maximum time for which the reference value will not be updated.

5.3.4 Fast Filter

The LTA has two filter values for each power mode. The filter value is selected based on the difference between the filtered counts and the LTA. The **LTA fast band** defines the delta threshold that needs to be reached for the LTA to use the fast beta filter values defined in the **filter beta** register.

5.4 Trigger-Level UI

The IQS7320A uses a trigger-level UI to detect key presses with various parameters. A channel activation event will occur to indicate that a key has been pressed when the output level of a certain channel has exceeded the trigger level of that channel. The level output of the trigger-level UI can be read from the **channel output level** registers. The trigger-level UI parameters are described below:

- **Max Delta**: Defines the range of delta count values which are divided by the number of thresholds selected. This parameter modifies the range of inductive measurement of a key. It is important to select a range in which the entire movement of a key can be observed. If the delta value of a channel exceeds this value, the output level will be equal to the number of thresholds.
- **Number of Thresholds**: Defines the number of levels into which the range of delta count values are divided. This parameter modifies the resolution of a key.
- **Trigger Level**: This defines the output level required for a channel to be activated, and for the relevant channel activation flag to be set. This parameter modifies the sensitivity of a key.

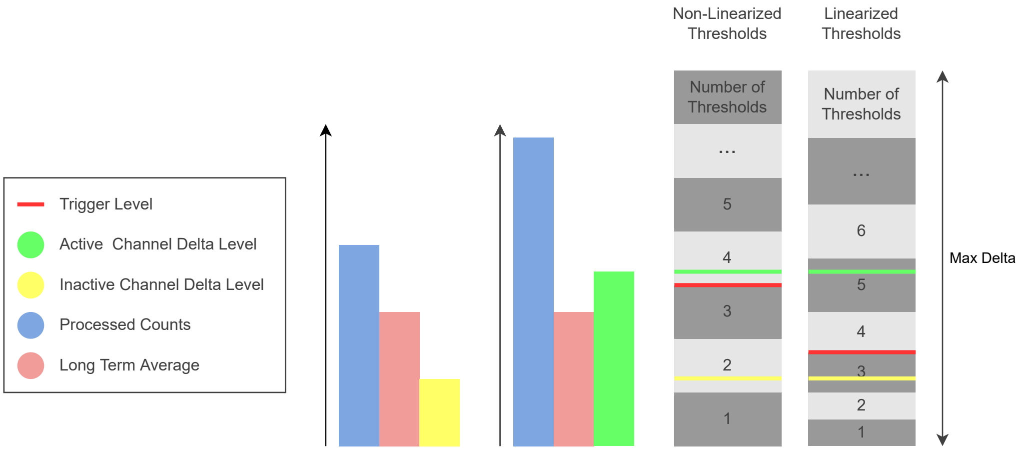


Figure 5.2: Trigger-Level UI Behaviour

5.4.1 Trigger-Level Hysteresis

The trigger-level hysteresis implementation prevents the trigger-level output from jittering between two levels. The hysteresis implementation on the IQS7320A will extend the lower threshold of the current output level. The **trigger-level hysteresis** parameter will define the size of hysteresis relative to the size of the threshold. This behavior is displayed in figure 5.3.

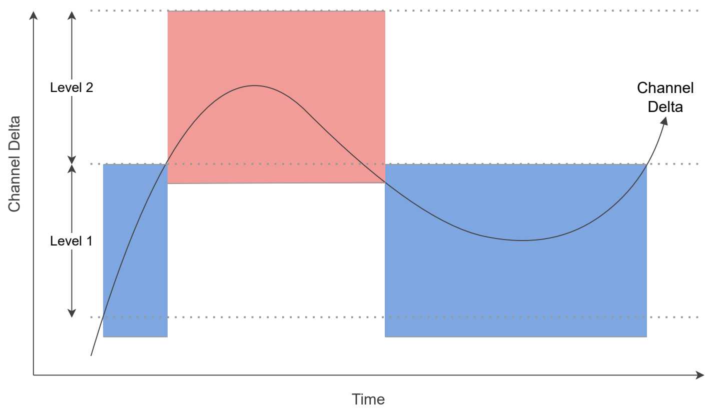


Figure 5.3: Trigger-Level Hysteresis

5.5 Automatic Tuning Implementation (ATI)

The ATI is a sophisticated technology implemented in the new ProxFusion® devices to allow optimal performance of the devices for a wide range of inductive sensing configurations without modification to external components. The ATI settings allow for the tuning of various parameters located in the **multipliers and dividers** and **compensation** registers.

For a detailed description of ATI, please contact Azoteq.

5.6 Automatic Re-ATI

5.6.1 Description

An automatic re-ATI will be performed when the reference value of a channel drifts outside the acceptable range around the ATI target. This range is defined by the ATI band defined in the **global channel settings**.

$$\text{Re-ATI Boundary} = \text{ATI Target} \pm \left(\frac{\text{ATI Target}}{2^{(\text{ATI Band} + 1)}}\right)$$

For example, assume that the ATI target is configured to 800 and the ATI band is set to 2, then :

$$\text{Re-ATI Boundary} = 800 \pm \frac{800}{2^3}$$

$$\text{Re-ATI Boundary} = 800 \pm 100$$

The ATI algorithm will be repeated under the following conditions:

$$\text{Reference} > 900 \text{ or Reference} < 700$$

The automatic re-ATI feature allows for easy and fast recovery from changes in the environment, adjustments to the selected working range, or an incorrect ATI event, such as when ATI was performed during user interaction with the sensor. The ATI algorithm executes in little time and goes unnoticed by the user.

5.6.2 ATI Error

The ATI error bit indicates that one or more of the channels did not achieve a count value within the selected ATI band after an ATI action has been performed. The ATI error bit can be observed in the **system status** register.

A re-ATI action will immediately be repeated if an ATI error occurs. This is done to recover from the ATI error as quickly as possible.

An ATI error would occur when any combination of ATI parameters would not produce the desired counts results for a channel. ATI errors may occur due to changes in the environment, defective hardware, or faulty hardware configuration.

6 Device Communication Methods

The IQS7320A has 2 communication methods

- Master-driven key scanning - For multiple devices
- Slave-driven I^{2}C streaming - For a single device

6.1 I^{2}C Streaming

In I^{2}C streaming (key scanning disabled) mode, the IQS7320A will not respond to GPIO toggle events and will instead allow the master device to communicate during I^{2}C ready periods, which occur after new samples have been measured. The device can be configured for I^{2}C streaming mode by setting the key scanning disable bit in the **system control** register.

The IQS7320A has an open-drain active low RDY signal to inform the master that updated data is available. The IQS7320A will pull the RDY line low to indicate that it has opened a communications window, or "RDY window", for the master to read the new updated data. Once the I^{2}C transactions have completed, and an I^{2}C stop condition is detected, the RDY line is released and the comms window is closed. The IQS7320A will then go to sleep or continue with a new sensing cycle. An example of I^{2}C communication is displayed in Section 10.4.

6.2 Key Scanning

The master device will toggle GPIO pins **S0**,**S1**,**D0**,**D1**,**P0** on the IQS7320A to achieve 3 possible states within the IQS7320A:

- Device and channel state scanning
- I^{2}C configuration mode
- Standby mode

6.2.1 Device and Channel State Scanning

The key scanning sequence to read the device reset flag and channel states on output pins D0 and D1 is displayed in Figure 6.1. The IQS7320A will accept key scanning input from pins S0 and S1 and produce output on pins D0 and D1. The sequence is explained below:

1. Falling edge on S0 and S1
2. Wait for setup time t_{setup} [i]
3. Rising edge on S0 or S1
4. Wait for setup time t_{setup}
5. Rising edge on S0 or S1
6. Wait for setup time t_{setup}
7. Rising edge on S0 or S1
8. Wait for latency time $t_{latency}$ [ii]
9. Keyscan routine completed

Table 6.1: Key Scanning Truth Table

Channel	LOW	HIGH
Device Reset State	Reset not acknowledged	Reset acknowledged
Channel 0 Key State	Channel trigger level not reached	Channel trigger level reached
Channel 1 Key State	Channel trigger level not reached	Channel trigger level reached
Channel 2 Key State	Channel trigger level not reached	Channel trigger level reached
Channel 3 Key State	Channel trigger level not reached	Channel trigger level reached

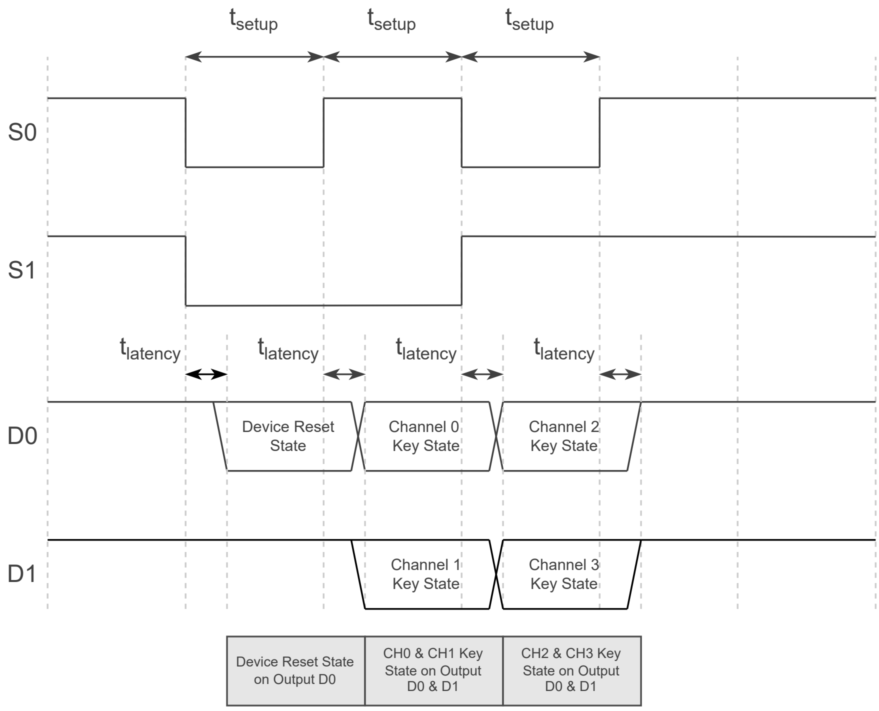


Figure 6.1: Key Scanning Sequence

[i] $t_{setup} = 20\mu s$
[ii] $t_{latency} <= 12\mu s$

The following code can be used as example code for key scanning:

```
void scan_all_keys()
{
    for(int j=0; j<NR_COLS; j++)
    {

        GPIO_PinWrite(col[j].s0, 0);            // set column Data Select 0 lines low
        GPIO_PinWrite(col[j].s1, 0);            // set column Data Select 1 lines low

        delay(t_setup);                         // wait for t_setup

        // Read row Device Reset State
        for(int k=0; k<NR_ROWS; k++)            // loop through rows
        {
            iqs_data[k][j].reset_occurred = GPIO_PinRead(row[k].d0); // check row reset
        }

        GPIO_PinWrite(col[j].s0, 1);            // set column Data Select 0 lines high

        delay(t_setup);                         // wait for t_setup

        // Read row Data Out lines for Channel 0 & 1 data
        for(int k=0; k<NR_ROWS; k++)
        {
            iqs_data[k][j].Trigger[0] = GPIO_PinRead(row[k].d0);  //Ch0
            iqs_data[k][j].Trigger[1] = GPIO_PinRead(row[k].d1);  //Ch1
        }

        GPIO_PinWrite(col[j].s0, 0);            // set column Data Select 0 lines low
        GPIO_PinWrite(col[j].s1, 1);            // set column Data Select 1 lines high

        delay(t_setup);                         // wait for t_setup

        // Read row Data Out lines for Channel 2 & 3 data
        for(int k=0; k<NR_ROWS; k++)
        {
            iqs_data[k][j].Trigger[2] = GPIO_PinRead(row[k].d0);  //Ch2
            iqs_data[k][j].Trigger[3] = GPIO_PinRead(row[k].d1);  //Ch3
        }

        GPIO_PinWrite(col[j].s0, 1);            // set column Data Select 0 lines high
    }
}
```

6.3 I^2C Configuration Mode

Multiple IQS7320A devices in a keyboard matrix share the same I^2C address and a specific routine is used to enable the I^2C functionality on a specific IC. The IQS7320A will accept key scanning input on pins S0, S1, and D1, and will not produce any output on GPIO pins during this key scanning sequence. The following sequence will allow the device to enter I^2C configuration mode:

1. Falling edge on S0 and S1
2. Wait for setup time t_{setup}
3. Rising edge on S0 and S1
4. Wait for setup time t_{setup}
5. Rising edge on D1
6. Wait for setup time t_{I2C_setup} [iii]
7. I^2C peripheral enabled

To disable the I^2C peripheral, the following sequence must be executed:

1. Wait for I2C to end
2. Falling edge on D1
3. Wait for setup time t_{I2C_setup}

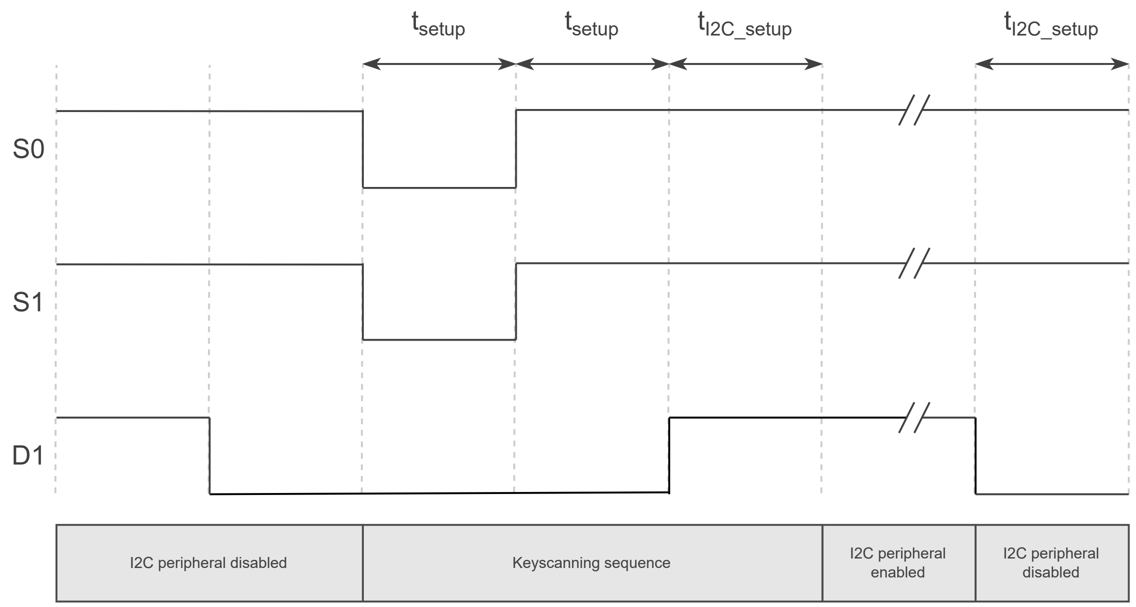


Figure 6.2: I^2C Configuration Process

[iii] $t_{I2C\_setup} = 30\mu s$

The example code below can be used as a reference to configure the I²C mode:

```
void write_settings_for_all_keys()
{
    for(int j=0; j<NR_COLS; j++)          // loop through columns
    {
        disable_all_i2c();
        for(int k=0; k<NR_ROWS; k++)      // loop through rows
        {
            enable_config_mode(k,j);
            I2C_read_write();             // perform I2C transactions
            disable_config_mode(k);
        }
    }
}

void disable_all_i2c(void)
{
    for(int k=0; k <NR_ROWS; k++)         // loop through rows
    {
        GPIO_PinWrite(row[k].d1, 1);      // set row Data Out 1 Line high
    }

    delay(t_i2c_setup);                   // wait for t_i2c_setup

    for(int k=0; k<NR_ROWS; k++)          // loop through rows
    {
        GPIO_PinWrite(row[k].d1, 0);      // set row Data Out 1 Lines low
    }

    delay(t_i2c_setup);                   // wait for t_i2c_setup
}

void enable_config_mode(uint8_t k, uint8_t j)
{
    GPIO_PinWrite(col[j].s0, 0);          // set column Data Select 0 line low
    GPIO_PinWrite(col[j].s1, 0);          // set column Data Select 1 line low

    delay(t_setup);                       // wait for t_setup

    GPIO_PinWrite(col[j].s0, 1);          // set column Data Select 0 line high
    GPIO_PinWrite(col[j].s1, 1);          // set column Data Select 1 line high

    delay(t_setup);                       // wait for t_setup

    GPIO_PinWrite(row[k].d1, 1);          // set row Data Out 1 line high

    delay(t_i2c_setup);                   // wait for t_i2c_setup
}

void disable_config_mode(uint8_t k)
{
    GPIO_PinWrite(row[k].d1, 0);      // set row Data Out 1 low

    delay(t_i2c_setup);               // wait for t_i2c_setup
}
```

6.4 Standby Mode

The IQS7320A will receive key scanning input on pins S0, S1, D0, D1, and P0 to enter and exit a low-power standby mode in which no analogue conversions occur.

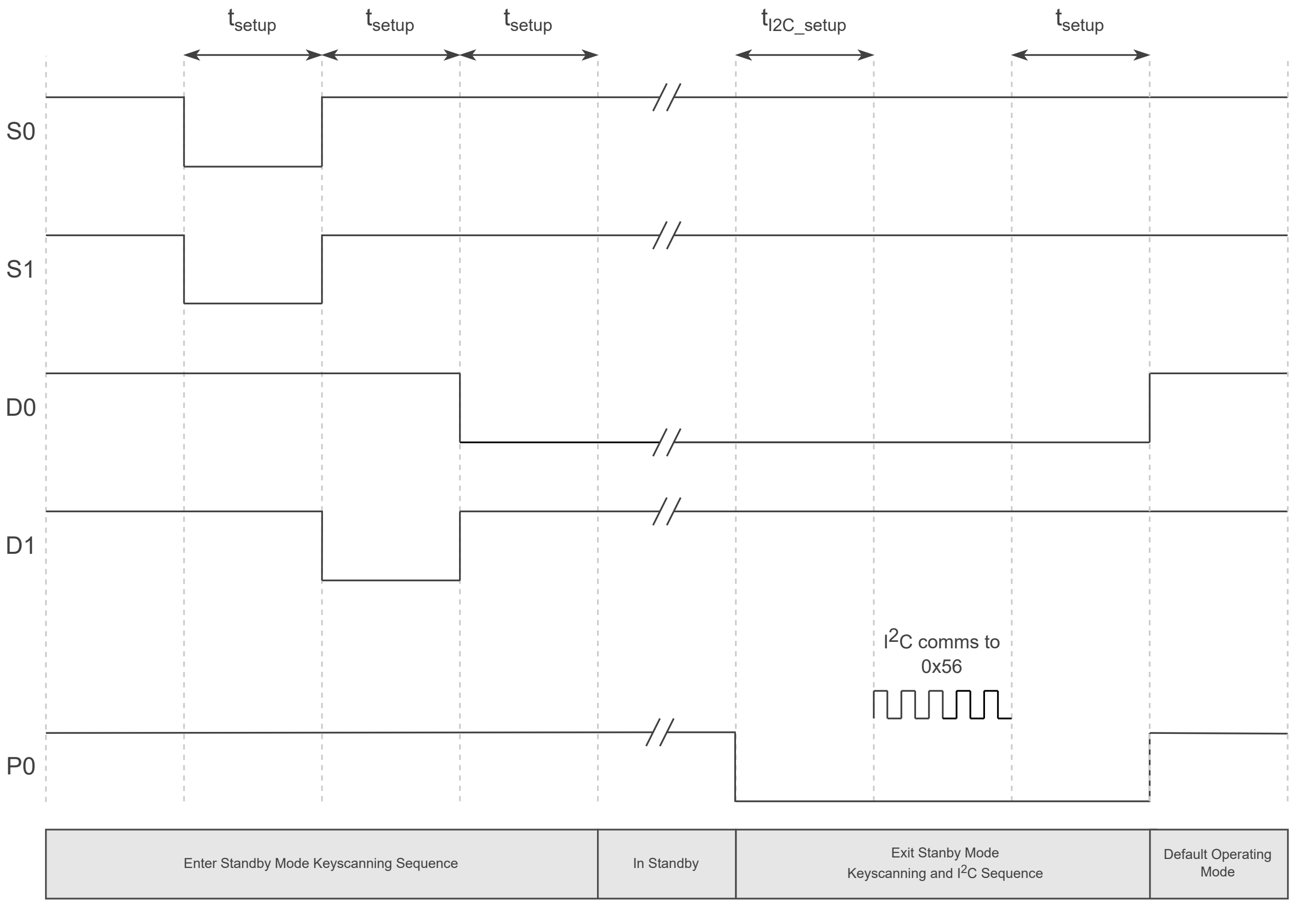


Figure 6.3: Standby Mode

Standby mode will be entered if the following sequence is followed:

1. Falling edge on S0 and S1
2. Wait for setup time t_{setup}
3. Rising edge on S0 and S1
4. Wait for setup time t_{setup}
5. D0 low and rising edge on D1
6. Wait for setup time t_{setup}
7. Device in Standby Mode

The sequence is visually shown in Figure 6.3. Standby mode will be exited if the following sequence is followed:

1. Falling edge on P0
2. Wait for setup time t_{I2C_setup}
3. Send device address via I^2C lines
4. Wait for setup time t_{setup}
5. Rising edge on P0
6. Device in default operational mode

The example code below can be used as a reference to enter and exit standby mode:

```
void enter_standby_mode(uint8_t k, uint8_t j)
{
    GPIO_PinWrite(col[j].s0, 0);  // set column Data Select 0 line low
    GPIO_PinWrite(col[j].s1, 0);  // set column Data Select 1 line low

    delay(t_setup);              // wait for t_setup

    GPIO_PinWrite(col[j].s0, 1);  // set column Data Select 0 line high
    GPIO_PinWrite(col[j].s1, 1);  // set column Data Select 1 line high
    GPIO_PinWrite(row[j].d1, 0);  // set row Data Out 1 line low

    delay(t_setup);              // wait for t_setup

    GPIO_PinWrite(row[j].d0, 0);  // set row Data Out 0 line low
    GPIO_PinWrite(row[j].d1, 1);  // set row Data Out 1 line high

    delay(t_setup);              // wait for t_setup
}

void exit_standby_mode(uint8_t k, uint8_t j)
{
    GPIO_PinWrite(p0, 0);        // set Sync line low

    delay(i2c_setup);              // wait for i2c_setup

    I2C_read();                  // perform I2C transaction

    delay(i2c_setup);            // wait for t_setup

    GPIO_PinWrite(p0, 1);        // set Sync line high

}
```

7 Timing Diagrams

7.1 Key Update

The key update period t_{update} is defined as the time required to perform analogue channel conversions on all channels $t_{analogue}$, plus the time required to digitally process the analogue conversions $t_{digital}$. Each device supports up to 4 individual keys which are assigned to CH0, CH1, CH2, and CH3. The analogue conversions on all 4 channels occurs over 2 cycle periods in which the conversions for CH0 and CH1 occur during cycle 1 and the conversions for CH2 and CH3 occur during cycle 2 (See Figure 5.1). Table 7.1 provides the device configuration for the setup used to measure the key update period as shown in Figure 7.1 and the respective timings for the key update routine are shown in Table 7.2.

Table 7.1: Key Update Device Configuration

Number of Cycles	: 2
Number of Channels	: 4
Conversion Frequency	: 9.0 MHz
ATI Target Counts	: 350
Interface Selection	: Asynchronous Conversion Mode

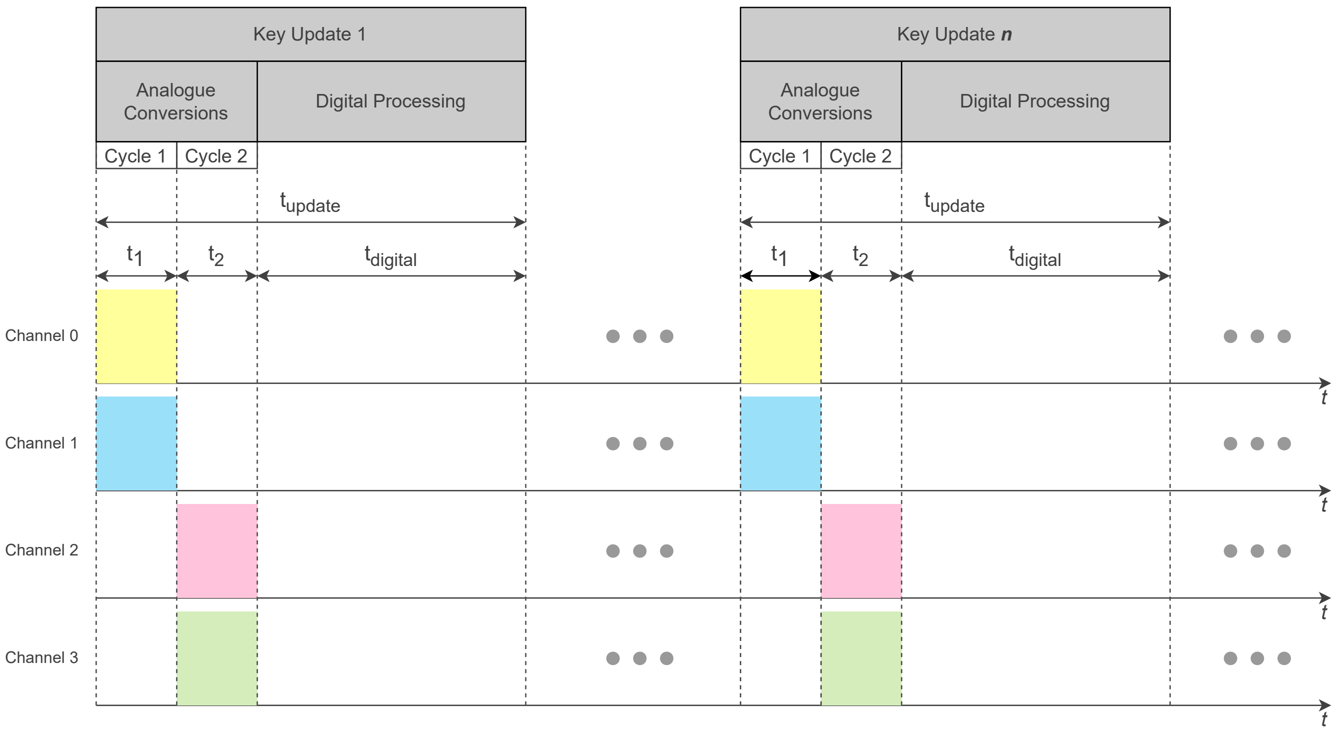


Figure 7.1: IQS7220A Key Update Routine Timing Diagram

Table 7.2: Key Update Routine Timings

Notation	Description	Average Time $[\mu s]$
t_1	Cycle 1 conversions (CH0 & CH1)	60
t_2	Cycle 2 conversions (CH2 & CH3)	60
$t_{analogue}$	Analogue conversion time for all 4 channels	120
$t_{digital}$	Digital processing time to update all 4 key outputs on D0 & D1	280
t_{update}	Total time required to perform analogue conversions and digital processing of all 4 channels	400

7.2 Key Scan Rate

From Figure 6.1, the minimum key scanning period t_{min_scan} is fixed as $4 \cdot t_{setup}$. The duration of t_{setup} depends on the selected device conversion mode.

- Synchronous conversion mode (Sync on key scan bit enabled. Address 0x50 bit 7)
  - $t_{setup} = 15\mu s$
  - $t_{min\_scan} = 4 \cdot t_{setup} = 60\mu s$
  - $t_{min\_scan\_update} = t_{min\_scan} + t_{update} = 60 + 400 = 460\mu s$
- Asynchronous conversion mode (Sync on key scan bit disabled. Address 0x50 bit 7).
  - $t_{setup} = 20\mu s$
  - $t_{min\_scan} = 4 \cdot t_{setup} = 80\mu s$
  - $t_{min\_scan\_update} = t_{min\_scan} + t_{update} = 80 + 400 = 480\mu s$

For efficient scanning of the key states, the key scan should only occur after the state of the keys has been updated. This means, t_{update} limits how fast the device can be scanned. From Table 7.2, t_{update} is larger than the fastest allowable scan time t_{min_scan}. Even though the device can be scanned every t_{min_scan}, it is the key update interval, t_{update}, that ultimately defines the fastest updated key scan interval $t_{min\_scan\_update}$.

7.3 Effect of Counts on Analogue Conversion Time

The time required to complete analogue conversions is dependent on the signal applied to a channel and the ATI parameters of the channel the signal is applied to. The resulting counts correlate to the amount of time required to produce a digital result for a given charge transfer frequency. The conversion period, update time, and minimum scan update time for a given ATI targer at the maximum charge transfer frequency of 9MHz is displayed in Table 7.3.

Table 7.3: Effect of Counts on Analogue Conversion Time

ATI Target	Conversion Time $[\mu s]$	Update Time $[\mu s]$	Minimum Scan Update Time $[\mu s]$
150	36	350	410
250	48	370	440
350	60	400	460
450	70	420	480
600	90	440	500
800	110	470	530

7.4 Synchronous Conversion Mode

A key press can occur at any time during device operation and the worst possible timing scenario shown in Figure 7.2 is considered for defining $t_{max\_scan\_update}$ in the synchronous conversion mode. This will occur when the user presses one of the 4 keys immediately after the analogue conversion cycle for that key is complete. Figure 7.2 shows the scenario where the user presses a cycle 1 key (CH0 or CH1) immediately after the completion of the cycle 1 analogue conversions.

The following sequence describes the worst-case scenario shown in Figure 7.2:

- The key press is missed by the cycle 1 analogue conversion during the 1st key update routine KU1
- The key press is measured and processed on the 2nd key update routine KU2
- The updated output state of the key press on pins D0 and D1 is read by the MCU during the keyscan period of the 3rd key update routine KU3
- $t_{max\_scan\_update}$ = 60us + 280us + 60us + 60us + 60us + 280us + 60us = 860us

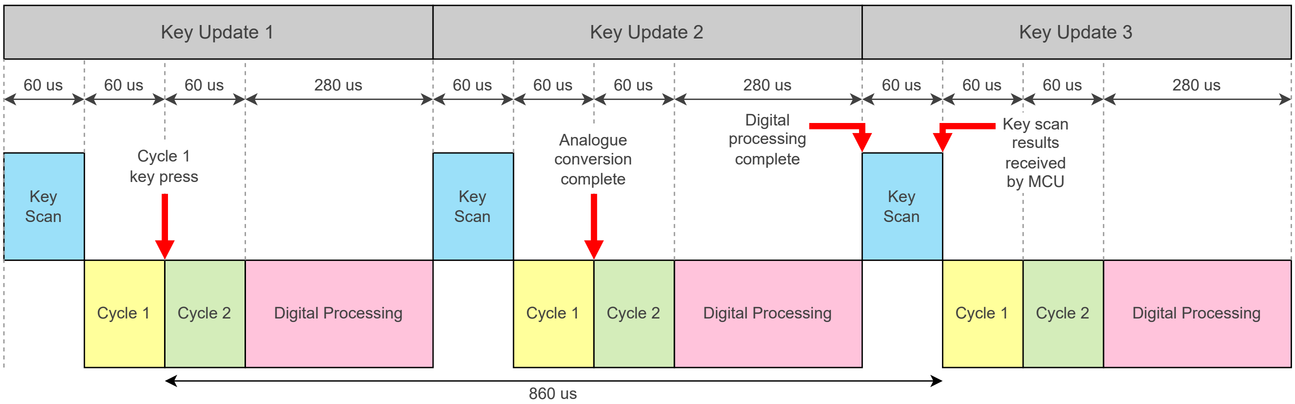


Figure 7.2: Synchronous mode worst case key update time

The number of columns, $n_{columns}$, dictates the number of keyscans required to read the status of all the keys. The time required to scan all the keys $t_{scan\_all\_columns}$ is given below (See Figure 7.3):

$$t_{scan\_all\_columns} = n_{columns} \cdot t_{min\_scan} \quad (1)$$

7.5 Multi-Column Key Scanning

For the synchronous conversion mode, the key scanning for all columns can occur in succession over the time interval $t_{min\_scan\_update}$ as shown in Figure 7.3.

The maximum number of columns n_{column_max} that can be scanned over the time interval $t_{min\_scan\_update}$, the time in which a newly sampled digital result can be produced, is given as:

$$n_{column\_max} = floor\left(\frac{t_{min\_scan\_update}}{t_{min\_scan}}\right) \tag{2}$$

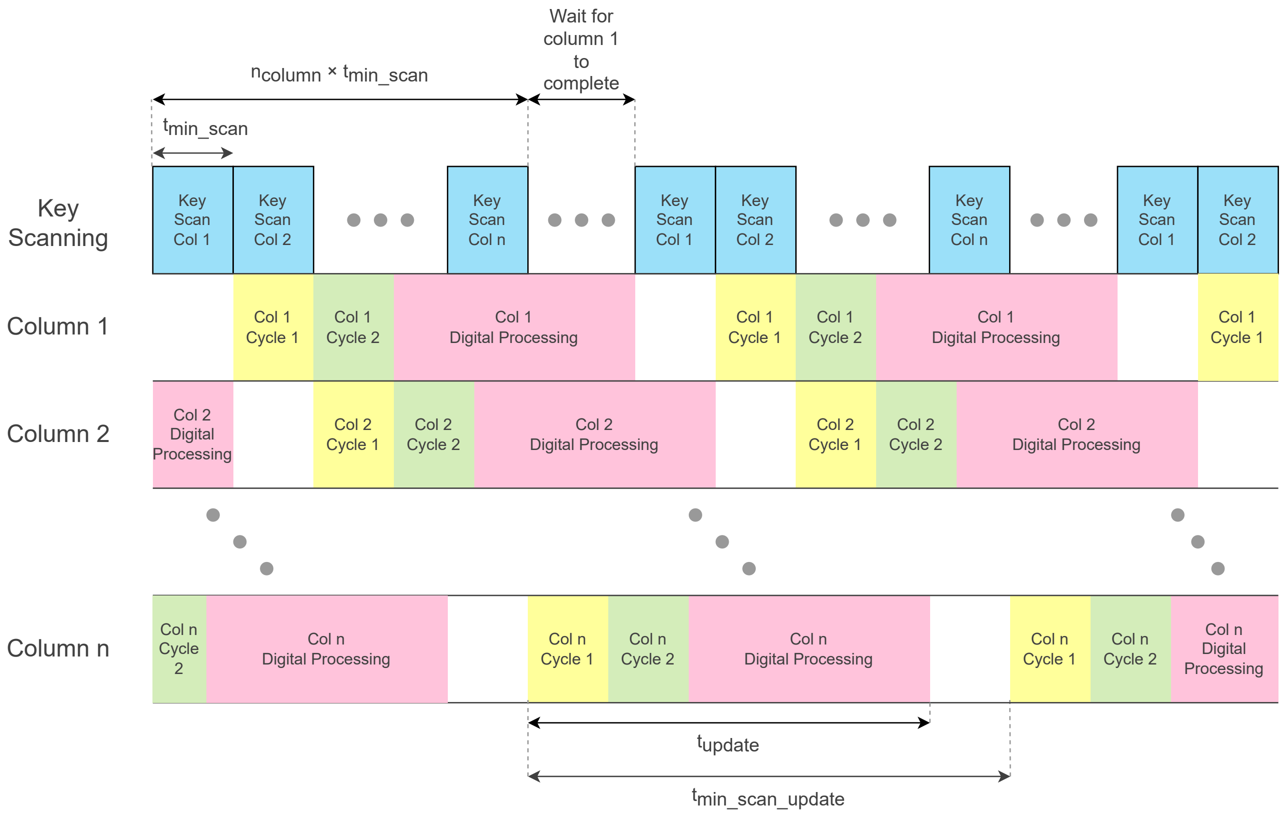


Figure 7.3: Synchronous conversion mode multi-column key scanning timings

7.6 Multiple Device Scanning

Each device requires the following communication lines:

- 2 Data Out lines
- 2 Data Select lines
- SYNC line (optional)
- I²C Clock and Data line

The following diagram visualizes a multiple device matrix:

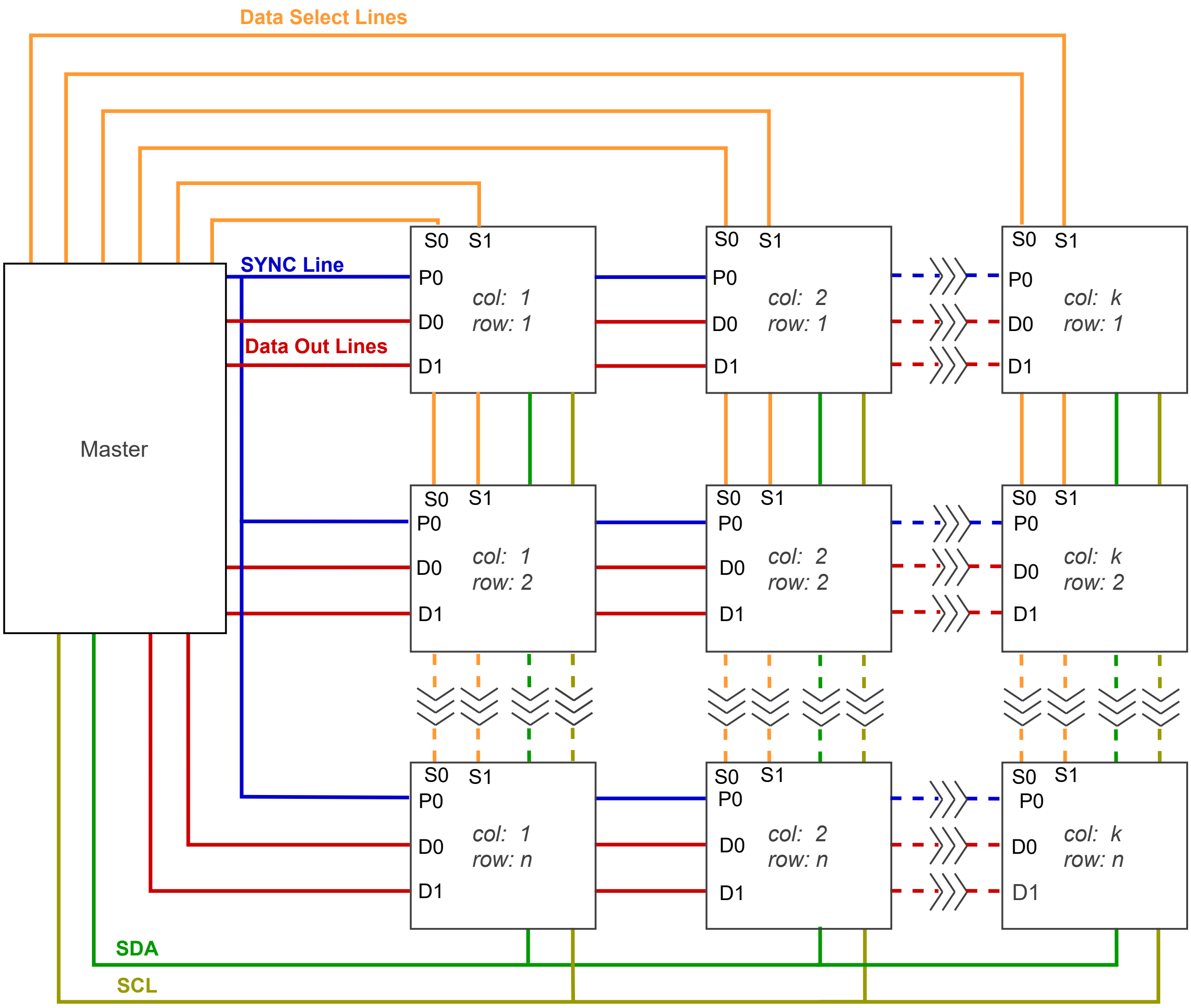


Figure 7.4: Multiple Device Scanning Matrix

The data select lines determine the keys that are simultaneously scanned during a column key scan sequence (Section 6.2.1). For a given number of keys n_{keys}, the number of IQS7220A devices required is given by:

$$n_{devices} = ceil\left(\frac{n_{keys}}{4}\right) \quad (3)$$

From Figure 7.4, the larger the number of rows the larger the number of keys that can be scanned during a given column key scan period t_{min_scan}. Thus, for fast response time, the matrix should be designed to maximize the number of rows for a given number of master IO pins n_{io_mcu}, (this excludes pins required for I²C and the SYNC lines). Since each device row has 2 IO lines and each device column also has 2 IO lines, then, n_{io_mcu} should always be rounded down to the nearest even number. The number of rows can then be calculated as:

$$n_{rows} = \frac{n_{io\_mcu} - \sqrt{(n_{io\_mcu})^2 - 4 * n_{keys}}}{2} \quad (4)$$

The number of columns for the rows calculated above is given by:

$$n_{columns} = \frac{n_{keys}}{4 \cdot n_{rows}} \quad (5)$$

The maximum n_{keys} that can be implemented for a given n_{io_mcu} can be determined by the steps below. These steps will give a square matrix output (or as close as possible to a square) for the n_{keys} that can be achieved with a set amount of n_{io_mcu}:

1. Determine the number of data select lines for the columns by dividing n_{io_mcu} with 2 and then rounding down to the nearest even number.
   - if n_{io_mcu} = 15
   - $n_{io\_columns} = round\_down\_even\left(\frac{15}{2}\right) = 6$
2. Divide $n_{io\_columns}$ by 2 to determine the number of columns.
   - $n_{columns} = \frac{n_{io\_columns}}{2}$
   - $n_{columns} = \frac{6}{2} = 3$
3. Subtract $n_{io\_columns}$ from n_{io_mcu} and round down to the nearest even number to get the number of IO lines for the rows.
   - $n_{io\_rows} = round\_down\_even(n_{io\_mcu} - n_{io\_columns})$
   - $n_{io\_rows} = round\_down\_even(15 - 6) = 8$
4. Divide n_{io_rows} by 2 to determine the number of rows
   - $n_{rows} = \frac{n_{io\_rows}}{2}$
   - $n_{rows} = \frac{8}{2} = 4$
5. Multiply $n_{columns}$ with n_{rows} to get the number of IQS7220A devices in the matrix
   - $n_{devices} = n_{columns} \times n_{rows}$
   - $n_{devices} = 3 \times 4 = 12$
6. Multiply $n_{devices}$ with 4 to get the number of keys that can be supported by the given n_{io_mcu}
   - $n_{keys} = n_{devices} \times 4$
   - $n_{keys} = 12 \times 4 = 48$
7. The number of MCU IO pins required for a given number of keys can be determined as follows:
   - $n_{io\_mcu} = round\_up\_even(\sqrt{4 \cdot n_{keys}})$
   - if n_{keys} = 104
   - $n_{io\_mcu} = round\_up\_even\left(\sqrt{4 \cdot 104}\right) = 22$

The table below provides a quick lookup for the minimum number of IOs required for a specified number of keys.

Table 7.4: Number Of Keys For A Given Number Of MCU IO Pins

Number of IO's n_{io_mcu}	**Number of Keys** n_{keys}	**Number of Columns** $n_{columns}$	**Number of Rows** n_{rows}	**Number of Devices** $n_{devices}$
4	4	1	1	1
6	8	1	2	2
8	16	2	2	4
10	24	2	3	6
12	36	3	3	9
14	48	3	4	12
16	64	4	4	16
18	80	4	5	20
20	100	5	5	25
22	120	5	6	30
24	144	6	6	36

8 Power Modes

The IQS7320A has 4 different power mode settings which can be selected in the **system control** register :

- Normal Power
- Low Power
- Ultra-Low Power
- Automatic Power Mode Switching

The effect of each power mode is defined by the selected communication method and if the synchronous mode of each method is selected. The recorded current consumption results are available in Section 3.4.

8.1 Automatic Power Mode

When the automatic power mode is selected the device will switch between power modes when no channel events are detected. The power mode timeout periods are adjustable values defined in the system settings register block (0x50 - 0x58). The device will return to normal power and reset the power mode timer when a channel event such as an activation event or filter halt event occurs.

8.2 I^{2}C Streaming Mode

The power consumption of the device in streaming mode is defined by the report rate of the selected power mode. A range of report rates can be selected for the different power modes with normal power mode being the fastest and ultra-low power mode being the slowest.

When the sync enable bit is set, the device will only do an analogue conversion and open a communication window after a falling edge has been detected on the sync line (GPIO6). This requires that the report rate of the current power mode must be greater than the rate at which falling edges occur on the sync line.

8.3 Key Scanning Mode

The normal power mode of the device in asynchronous key scanning mode will produce the fastest sampling rate available on the device since no time is spent awaiting I^{2}C communication or applying energy-saving measures. A low-power mode is also available with the asynchronous key scanning communication mode in which a parameter can be defined for setting the period between analogue conversions and for defining the rate at which newly measured samples can be expected.

If the device is in synchronous key scanning mode with the wait for key scan bit set in the **system control** register, the device will require a key scan event to start analogue conversions and calculate digital results. The current usage of the device in synchronous key scanning mode is defined by both the report rate of the current power mode, and the rate at which key scanning events occur.

9 Additional Features

9.1 Syncing Mode

The syncing mode of the IQS7320A allows the device to only take samples when an interrupt is triggered on a falling edge of the sync line located on GPIO6. A single conversion is then executed after the interrupt has been triggered. After a conversion has been completed, the IQS7320A will toggle the RDY line low to indicate the data from the IC is available. This feature allows the master device to control the rate at which conversions occur and samples are produced.

Syncing mode can be enabled by setting the sync enable bit in the **system control** register. The device will still respond to forced I^{2}C communications as explained in Section 10.9.3.

9.2 Watchdog Timer (WDT)

A software watchdog timer is implemented to improve system reliability.

The working of this timer is as follows:

- This timer is reset at a strategic point in the main loop.
- Failing to reset this timer will cause the appropriate ISR (interrupt service routine) to run.
- This ISR performs a software-triggered POR (Power on Reset).
- The device will reboot.
- The WDT timeout register determines the period in milliseconds before the device will reset.
- Ensure that the watchdog timeout period is greater than the I^{2}C timeout period.

9.3 RF Immunity

The IQS7320A has immunity to high power RF noise. To improve the RF immunity, extra decoupling capacitors are suggested on VREG and VDD.

Place a 100pF in parallel with the 2.2μF ceramic on VREG. Place a 4.7μF ceramic on VDD. All decoupling capacitors should be placed as close as possible to the VDD and VREG pads.

If needed, series resistors can be added to Rx electrodes to reduce RF coupling into the sending pads. Normally these are in the range of 470Ω-1kΩ. PCB ground planes also improve noise immunity.

9.4 Reset

9.4.1 Reset Indication

After a reset, the **device reset** bit will be set by the system to indicate the reset event occurred. This bit will be cleared when the master sets the **ACK reset** bit. If the device reset bit is set again, the master will know a reset has occurred on the device.

9.4.2 Software Reset

The IQS7320A can be reset by means of an I^{2}C command. The software reset bit in the **system control** register must be set for the device to reset.

10 I²C Interface

10.1 I²C Module Specification

The device supports a standard two wire I²C interface with the addition of an RDY (ready interrupt) line. The communications interface of the IQS7320A supports the following:

- *Fast-mode-plus* standard I²C up to 1MHz.
- I²C peripheral enable on IO config.
- The provided interrupt line (RDY) is an open-drain active low implementation and indicates a communication window.

The IQS7320A implements 8-bit addressing with 16-bit values stored at each address. Two consecutive reads/writes are required in this memory map structure. The two bytes at each address will be referred to as "byte 0" (least significant byte) and "byte 1" (most significant byte).

10.2 I²C Address

The default 7-bit device address is 0x44 ('01000100'). The full address byte will thus be 0x88 (read) or 0x89 (write).
Other address options exist on special request. Please contact Azoteq.

10.3 I³C Compatibility

This device is not compatible with an I³C bus due to clock stretching allowed for data retrieval.

10.4 Data

The data is 16-bit words, meaning that each address obtains 2 bytes of data. For example, address 0x10 will provide two bytes, then the next two bytes read will be from address 0x11. The 16-bit data is sent in little-endian byte order (least significant byte first).

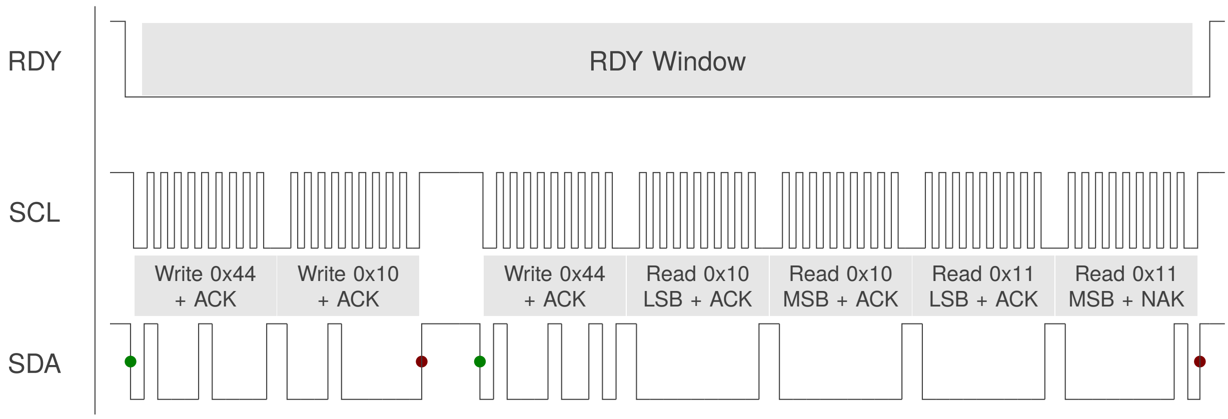


Figure 10.1: Example of reading data over I²C

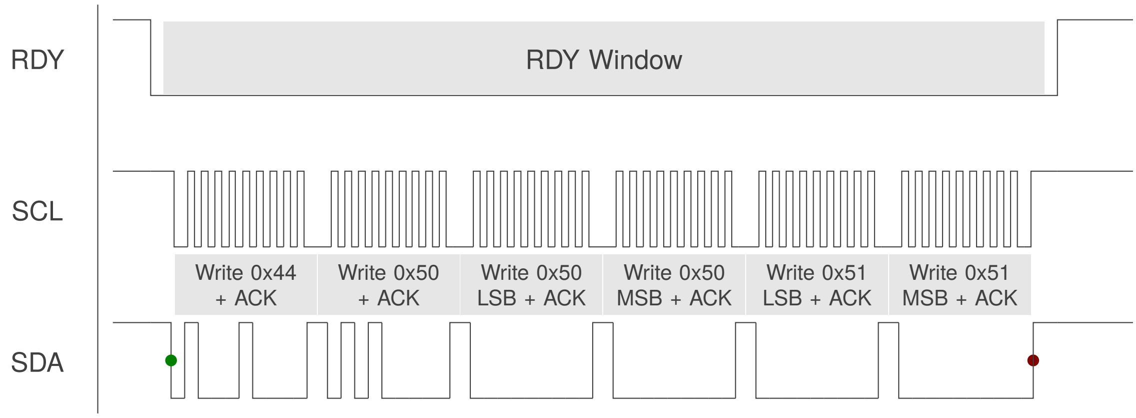


Figure 10.2: Example of writing data over I^2C

The .h file generated by the GUI will display the start address of each block of data, with each address containing 2 bytes. The data of all the addresses can be written consecutively in a single block of data, as shown in Figure 10.2. An example of the .h file exported by the GUI is shown in below.

```
/* Change the Global Device Settings */
/* Memory Map Position 0x30 - 0x3F */   // Shows starting address
#define GLOBAL_CHANNEL_SETTINGS_0        0xA8  // LSB on 0x30
#define GLOBAL_CHANNEL_SETTINGS_1        0x52  // MSB on 0x30
#define LTA_FILTER_SETTINGS_0            0x79  // LSB on 0x31
#define LTA_FILTER_SETTINGS_1            0xBD  // MSB on 0x31
...
```

10.5 I^2C Timeout

If the communication window is not serviced within the I^2C timeout period (in milliseconds), the session is ended (RDY goes HIGH), and processing continues as normal. This allows the system to continue and keep reference values up to date even if the master is not responsive, however the corresponding data was missed/lost, and this should be avoided. The default I^2C timeout period is set to 5ms, and can be set in address 0x65 under the **engineering settings** register.

10.6 Terminate Communication

A standard I^2C STOP ends the current communication window.

10.7 RDY/IRQ

The communication has an open-drain active-LOW RDY signal to inform the master that updated data is available. It is optimal for the master to use this as an interrupt input and obtain the data accordingly. It is also useful to allow the master MCU to enter low-power/sleep allowing wake-up from the touch device when user presence is detected. It is recommended that the RDY be placed on an interrupt-on-pin-change input on the master.

10.8 Invalid Communications Return

The device will give an invalid communication response (0xEE) under the following conditions:

- The host is trying to read from a memory map register that does not exist.
- The host is trying to read from the device outside of a communication window (i.e. while RDY = high)

10.9 I²C Interface

The IQS7320A has 2 I²C interface options as described in the sections below.

10.9.1 I²C Streaming

The device will periodically open communication windows, independent of the data sampled by the device. The period between communication windows is defined by the report rate of the active power mode.

10.9.2 Sync Mode

The device will only open communication windows when a falling edge is detected on the sync line on pin P0. Prior to opening the communication window, the device will do a single conversion on each channel.

10.9.3 Force Communication

In streaming mode, the IQS7320A I²C will provide ready windows at intervals specified in the power mode report rate. Ideally, communication with the IQS7320A should only be initiated in a ready window but a communication request described in figure 10.3 below, will force a Ready window to open. In event mode, ready windows are only provided when an event is reported and a ready window must be requested to write or read settings outside of this window. The minimum and maximum time between the communication request and the opening of a ready window (t_{wait}), is application specific, but the average values are $0.1\text{ms} \leq t_{wait} \leq 5\text{ms}$ [i].

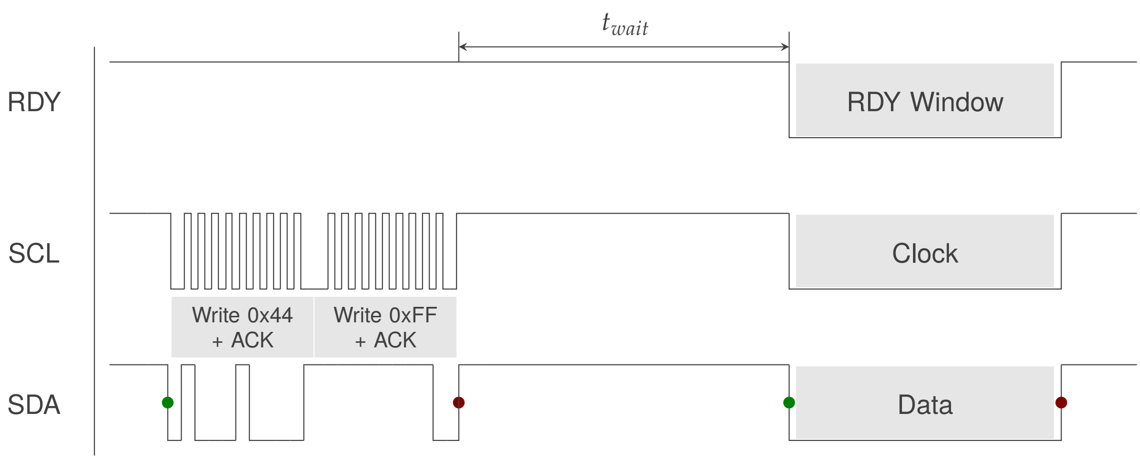


Figure 10.3: Force Communication Sequence

[i] Please contact Azoteq for an application-specific value of t_{wait}

11 I²C Memory Map - Register Descriptions

See Appendix A for a more detailed description of registers and bit definitions

Address	Data (16bit)	Notes
0x00 - 0x09	Version details	See Table A.1
Read Only	**System status and counts**	
0x10	System Status	See Table A.2
0x11	Events	See Table A.3
0x12	Channel 0 and 1 Output Levels	See Table A.4
0x13	Channel 2 and 3 Output Levels	See Table A.5
0x14	Channel 0 Processed Counts	Unsigned 16-bit value
0x15	Channel 1 Processed Counts	
0x16	Channel 2 Processed Counts	
0x17	Channel 3 Processed Counts	
0x18	Channel 0 LTA	
0x19	Channel 1 LTA	
0x1A	Channel 2 LTA	
0x1B	Channel 3 LTA	
0x1C	Channel 0 Delta	
0x1D	Channel 1 Delta	
0x1E	Channel 2 Delta	
0x1F	Channel 3 Delta	
Read-Write	**Channel Settings**	
0x30	Global Channel Settings	See Table A.6
0x31	LTA Filter Settings	See Table A.7
0x32	Global ATI Base	Unsigned 16-bit value
0x33	Global ATI Target	Unsigned 16-bit value
0x34	Global Detection Settings	See Table A.8
0x35	Global Detection Hysteresis	See Table A.9
0x36	Channel 0 Max Delta	Unsigned 16-bit value
0x37	Channel 1 Max Delta	
0x38	Channel 2 Max Delta	
0x39	Channel 3 Max Delta	
0x3A	Detection Timeouts	See Table A.10
0x3B	I²C and Key Scan Timeouts	See Table A.11
0x3C	Channel 0 Rx/Tx selection	See Table A.12
0x3D	Channel 1 Rx/Tx selection	
0x3E	Channel 2 Rx/Tx selection	
0x3F	Channel 3 Rx/Tx selection	
Read-Write	**Detection Settings**	
0x40	Channel 0 & Channel 1 Trigger level	See Table A.4
0x41	Channel 2 & Channel 3 Trigger level	See Table A.5
0x42	Channel 0 Measured Max Delta	Unsigned 16-bit value
0x43	Channel 1 Measured Max Delta	
0x44	Channel 2 Measured Max Delta	
0x45	Channel 3 Measured Max Delta	
Read-Write	**System Settings**	
0x50	System Control	See Table A.13
0x51	Channel Control	See Table A.14

0x52	Key Scan Normal Power Timeout	Unsigned 16-bit value (ms)
0x53	Key Scan Low Power Report Rate	
0x54	Streaming Normal Power Report Rate	
0x55	Streaming Normal Power Timeout	
0x56	Streaming Low Power Report Rate	
0x57	Streaming Low Power Timeout	
0x58	Streaming Ultra Low Power Report Rate	
Read-Write	**Engineering Settings**	
0x60	Charge Transfer Frequency Settings	See Table A.15
0x61	Hardware Settings 0	See Table A.16
0x62	Hardware Settings 1	See Table A.17
0x63	WDT Timeout	Unsigned 16-bit value (ms)
0x64	RF Blocking Settings	See Table A.18
0x65	I^{2}C Timeout	Unsigned 16-bit value (ms)
Read-Write	**ATI Parameters**	
0x70	Channel 0 Fine and Coarse Multipliers	See Table A.19
0x71	Channel 0 ATI Compensation	See Table A.20
0x72	Channel 1 Fine and Coarse Multipliers	See Table A.19
0x73	Channel 1 ATI Compensation	See Table A.20
0x74	Channel 2 Fine and Coarse Multipliers	See Table A.19
0x75	Channel 2 ATI Compensation	See Table A.20
0x76	Channel 3 Fine and Coarse Multipliers	See Table A.19
0x77	Channel 3 ATI Compensation	See Table A.20

12 Implementation and Layout

12.1 Layout Fundamentals

NOTE
Information in the following Applications section is not part of the Azoteq component specification, and Azoteq does not warrant its accuracy or completeness. Azoteq's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

12.1.1 Power Supply Decoupling

Azoteq recommends connecting a combination of a 4.7 µF plus a 100 pF low-ESR ceramic decoupling capacitor between the VDD and VSS pins. Higher-value capacitors may be used but can impact supply rail ramp-up time. Decoupling capacitors must be placed as close as possible to the pins that they decouple (within a few millimetres).

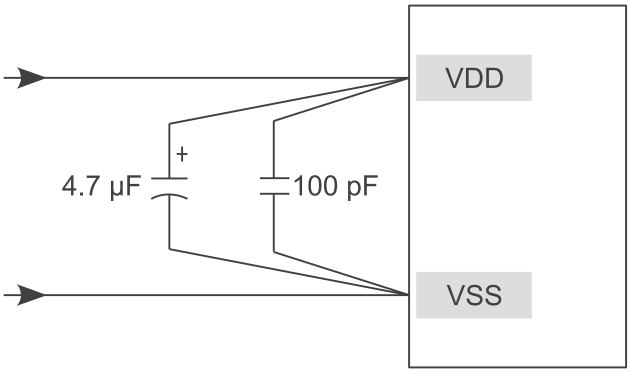


Figure 12.1: Recommended Power Supply Decoupling

12.1.2 VREG

The VREG pin requires a 2.2 µF capacitor to regulate the LDO internal to the device. This capacitor must be placed as close as possible to the microcontroller. The figure below shows an example layout where the capacitor is placed close to the IC.

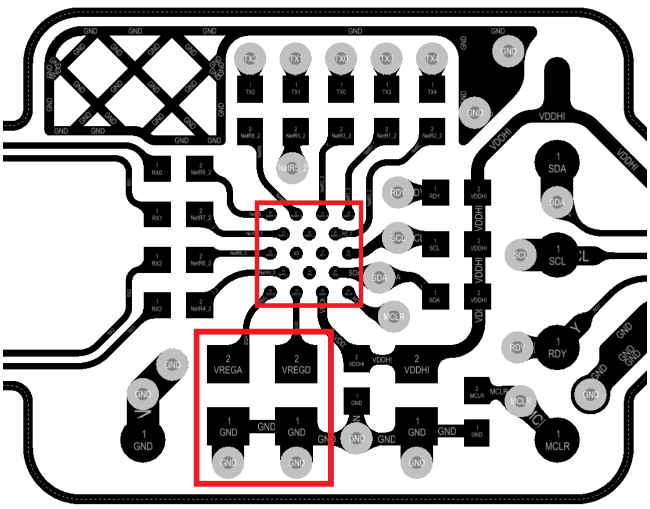


Figure 12.2: VREG Capacitor Placement Close to IC

13 Ordering Information

13.1 Ordering Code

IQS7320A zzz ppb

IC NAME	IQS7320A	=	IQS7320A	
POWER-ON CONFIGURATION	zzz	=	001	Reserved
			101	8 button self capacitance startup, configurable via I^2C.
			102	8 button self capacitance startup, configurable via I^2C. I^2C address = 0x45
PACKAGE TYPE	pp	=	CS	WLCSP-18 package
			QN	QFN-20 package
BULK PACKAGING	b	=	R	WLCSP-18 Reel (3000pcs/reel)
				QFN-20 Reel (2000pcs/reel)

Figure 13.1: Order Code Description

13.2 Top Marking

13.2.1 WLCSP18 Package

IQS 7320A pppxx •	Product Name ppp = product code xx = batchcode

13.2.2 QFN20 Package Marking Option 1

• IQS 7320A pppxx	Product Name ppp = product code xx = batchcode

13.2.3 QFN20 Package Marking Option 2

• IQS 722xy pppxx	Product Name ppp = product code xx = batchcode

14 Package Specification

14.1 Package Outline Description – QFN20 (QFR)

This package outline is specific to order codes ending in *QFR*.

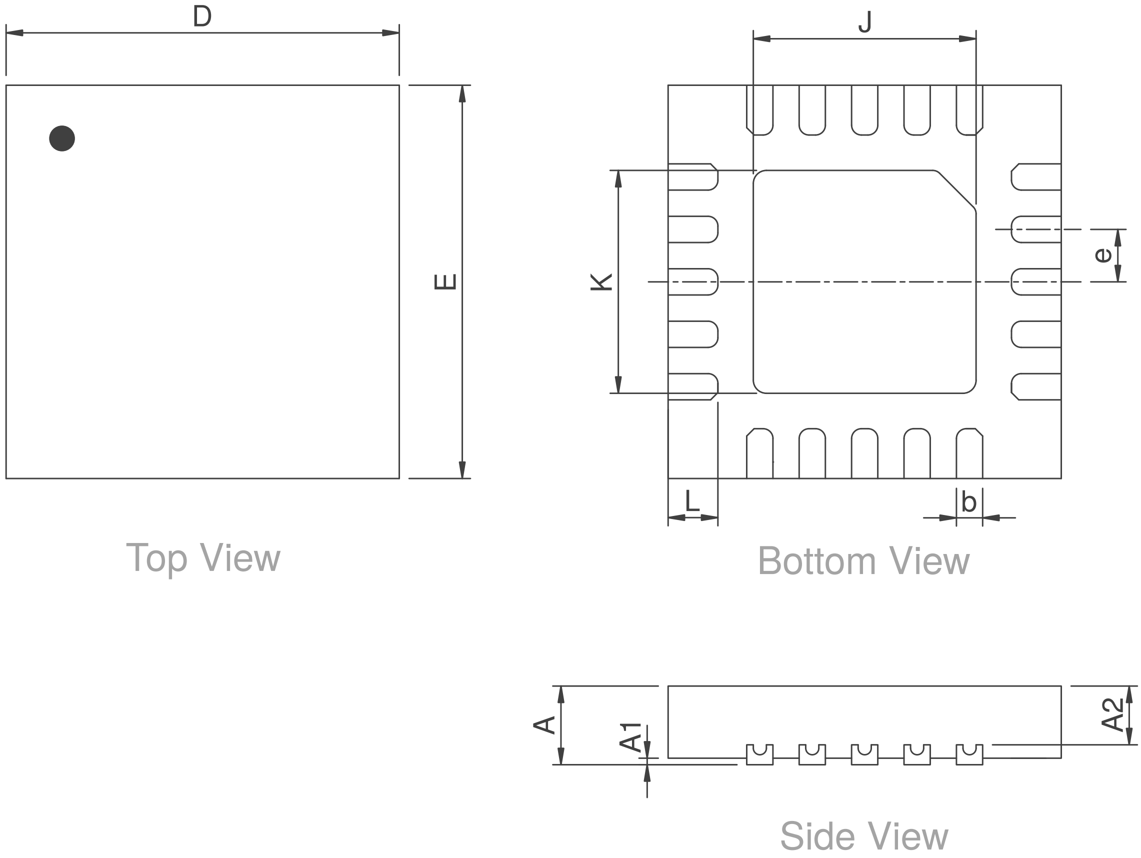


Figure 14.1: QFN (3x3)-20 (QFR) Package Outline Visual Description

Table 14.1: QFN (3x3)-20 Package Outline Visual Description

Dimension	[mm]	Dimension	[mm]
A	0.55 ± 0.05	E	3
A1	0.035 ± 0.05	e	0.4
A2	0.3	J	1.7 ± 0.1
b	0.2 ± 0.05	K	1.7 ± 0.1
D	3	L	0.3 ± 0.05

14.2 Package Outline Description – QFN20 (QNR)

This package outline is specific to order codes ending in *QNR*.

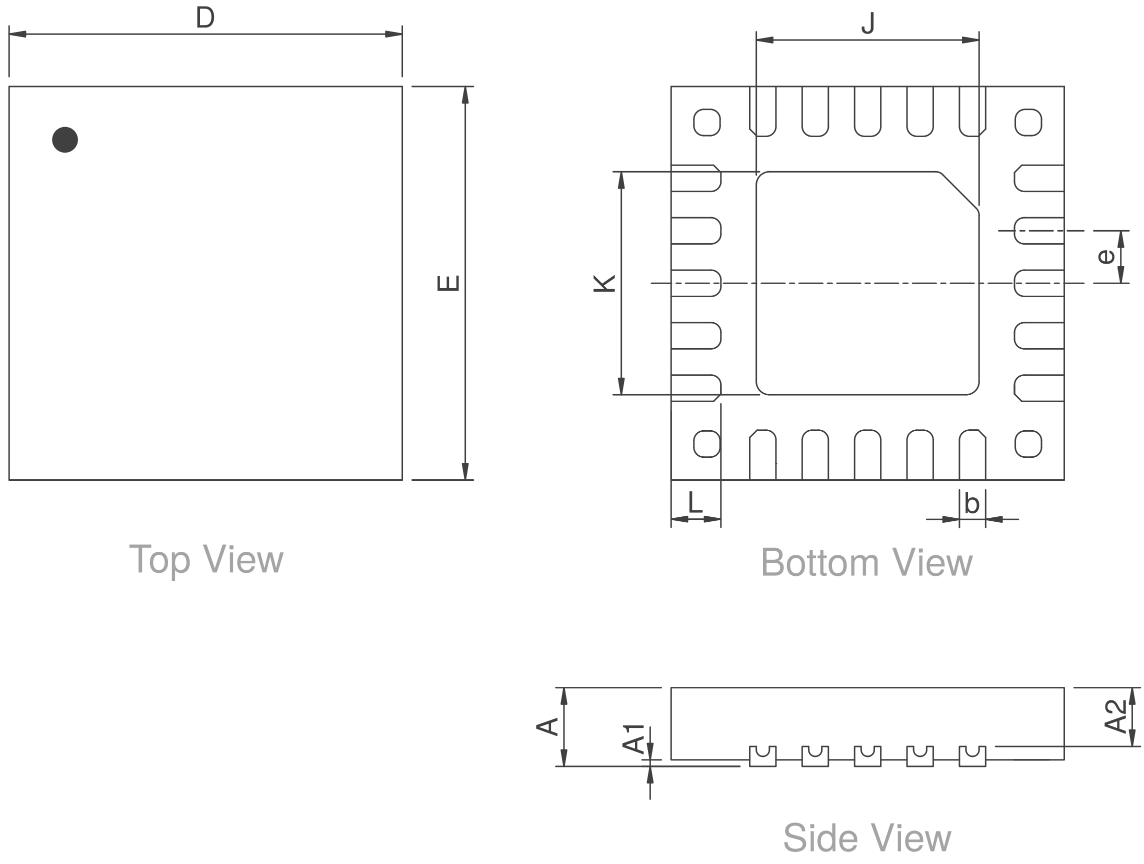
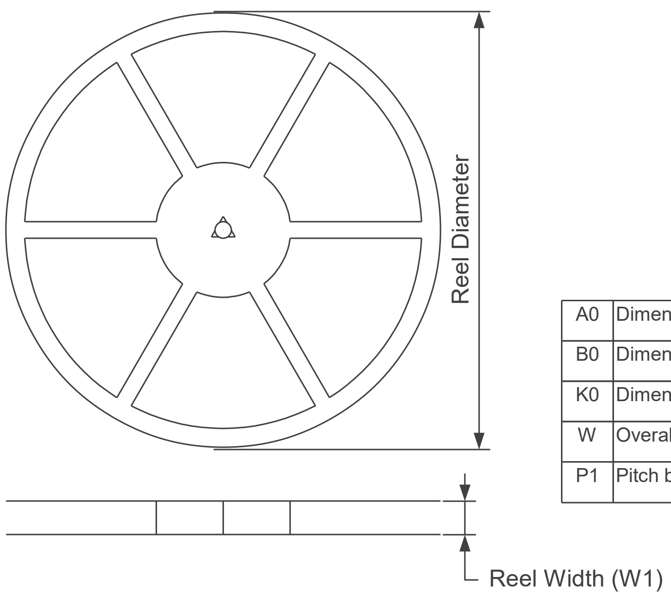


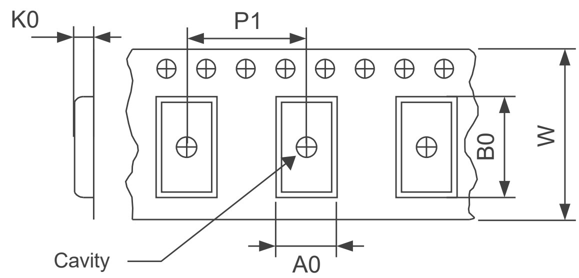
Figure 14.2: QFN (3x3)-20 (QNR) Package Outline Visual Description

Table 14.2: QFN (3x3)-20 Package Outline Visual Description

Dimension	[mm]	Dimension	[mm]
A	0.55 ± 0.05	E	3
A1	0.035 ± 0.05	e	0.4
A2	0.3	J	1.7 ± 0.1
b	0.2 ± 0.05	K	1.7 ± 0.1
D	3	L	0.38 ± 0.05

14.3 Tape and Reel Specifications





A0	Dimension designed to accommodate the component width
B0	Dimension designed to accommodate the component length
K0	Dimension designed to accommodate the component thickness
W	Overall width of the carrier tape
P1	Pitch between successive cavity centers

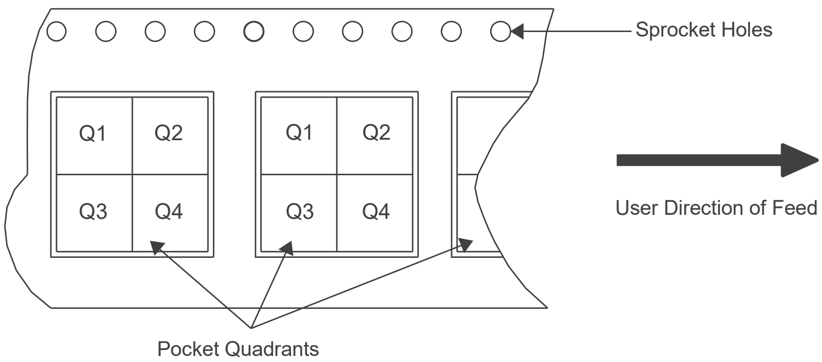


Figure 14.3: Tape and Reel Specification

Table 14.3: Tape and Reel Specifications

Package Type	Pins	Reel Diameter (mm)	Reel Width (mm)	A0 (mm)	B0 (mm)	K0 (mm)	P1 (mm)	W (mm)	Pin1 Quadrant
QFN20	20	180	12.4	3.3	3.3	0.8	8	12	Q2

14.4 Moisture Sensitivity Levels

Table 14.4: Moisture Sensitivity Levels

Package	MSL
QFN20	1

14.5 Reflow Specifications

Contact Azoteq

A Memory Map Descriptions

Table A.1: Version Information

Address	Category	Name	Value		
0x00	Application Version Info	Product Number	1729		
0x01		Major Version	1 (for order code 001)	2	2
0x02		Minor Version	13	6	23/27[i]

Table A.2: System Status

System Status (0x10)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved				CH0 RF Block	CH1 RF Block	CH2 RF Block	CH3 RF Block	Reserved		Power Mode		Device Reset	In ATI	Event Occurred	ATI Error

- Bit 11: **Channel 3 RF Block**
  - 0: The channel has not detected RF noise.
  - 1: The RF block check has been triggered due to noise.
- Bit 10: **Channel 2 RF Block**
  - 0: The channel has not detected RF noise.
  - 1: The RF block check has been triggered due to noise.
- Bit 9: **Channel 1 RF Block**
  - 0: The channel has not detected RF noise.
  - 1: The RF block check has been triggered due to noise.
- Bit 8: **Channel 0 RF Block**
  - 0: The channel has not detected RF noise.
  - 1: The RF block check has been triggered due to noise.
- Bit 4-5: **Power Mode**
  - 00: Normal Power
  - 01: Low Power
  - 10: Ultra-Low Power
- Bit 3: **Device Reset**
  - 0: Device reset is acknowledged
  - 1: Device reset is not acknowledged
- Bit 2: **In ATI**
  - 0: Device is not actively setting ATI parameters
  - 1: Device is actively setting ATI parameters to reach a target
- Bit 1: **Event Occurred**
  - 0: No event occurred
  - 1: An event has occurred
- Bit 0: **ATI Error**
  - 0: Device is not in ATI error state
  - 1: Device is in ATI error state - A channel cannot reach the selected target value

Table A.3: Events

Events (0x11)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved								CH3 Activation	CH2 Activation	CH1 Activation	CH0 Activation	CH3 Filter Halt	CH2 Filter Halt	CH1 Filter Halt	CH0 Filter Halt

- Bit 7: **CH3 Activation**
  - 0: The channel's activation threshold has not been reached.
  - 1: The channel's activation threshold has been reached.
- Bit 6: **CH2 Activation**
  - 0: The channel's activation threshold has not been reached.
  - 1: The channel's activation threshold has been reached.

[i] Please refer to PCN-PR560-IQS7320A Product Change note v1.0

- Bit 5: **CH1 Activation**
  - 0: The channel's activation threshold has not been reached.
  - 1: The channel's activation threshold has been reached.
- Bit 4: **CH0 Activation**
  - 0: The channel's activation threshold has not been reached.
  - 1: The channel's activation threshold has been reached.
- Bit 3: **CH3 Filter Halt**
  - 0: The channel's filter-halt threshold has not been reached.
  - 1: The channel's filter-halt threshold has been reached.
- Bit 2: **CH2 Filter Halt**
  - 0: The channel's filter-halt threshold has not been reached.
  - 1: The channel's filter-halt threshold has been reached.
- Bit 1: **CH1 Filter Halt**
  - 0: The channel's filter-halt threshold has not been reached.
  - 1: The channel's filter-halt threshold has been reached.
- Bit 0: **CH0 Filter Halt**
  - 0: The channel's filter-halt threshold has not been reached.
  - 1: The channel's filter-halt threshold has been reached.

Table A.4: Channel 0&1 Level Output

Channel 0&1 Level Output (0x12)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CH1 Level Output								CH0 Level Output							

- Bit 8-15: **CH1 Level Output** The current output level of the channel (dependent on the max delta, number of thresholds and the channel's delta.)
- Bit 0-7: **CH0 Level Output** The current output level of the channel (dependent on the max delta, number of thresholds and the channel's delta.)

Table A.5: Channel 2&3 Level Output

Channel 2&3 Level Output (0x13)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CH3 Level Output								CH2 Level Output							

- Bit 8-15: **CH3 Level Output** The current output level of the channel (dependent on the max delta, number of thresholds and the channel's delta.)
- Bit 0-7: **CH2 Level Output** The current output level of the channel (dependent on the max delta, number of thresholds and the channel's delta.)

Table A.6: Global Channel Settings

Global Channel Settings (0x30)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Res-erved	ATI Mode			Num Filtered Counts		ATI Band		Fast LTA Threshold				Linear-ize Thresh-olds	Temp-erature Track-ing	RF Filter Block	Global Filter Halt

- Bit 12-14: **ATI Mode**
  - 000: ATI disabled
  - 001: Compensation only
  - 010: ATI from compensation divider
  - 011: ATI from fine fractional divider
  - 100: ATI from coarse fractional divider
  - 101: Full ATI
- Bit 10-11: **Number of Filtered Counts**
  - 00: No counts are averaged (raw counts)
  - 01: Past two raw samples are averaged
  - 10: Past three raw samples are averaged
  - 11: Past four raw samples are averaged

- Bit 8-9: **ATI Band**
  - 00: No ATI band checking
  - 01: ATI band = ATI target/4
  - 10: ATI band = ATI target/8
  - 11: ATI band = ATI target/16
- Bit 4-7: **Fast LTA Band**
  - 4-bit value, the delta that the counts need to reach in the opposite direction of event detection before the fast LTA beta is used.
- Bit 3: **Linearized Thresholds**
  - 0: Channel level thresholds are not linearized.
  - 1: Channel level thresholds are linearized.
- Bit 2: **Temperature Tracking**
  - 0: Delta adjustment temperature tracking UI is disabled.
  - 1: Delta adjustment temperature tracking UI is enabled.
- Bit 1: **RF Filter Block**
  - 0: RF filter block UI is disabled.
  - 1: RF filter block UI is enabled.
- Bit 0: **Global Filter Halt**
  - 0: Each channel determines its filter halt condition.
  - 1: If a single channel is in a filter halt condition, all other channels' LTA will be halted.

Table A.7: LTA Filter Betas

Filter Betas (0x31)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
LTA Low Power Fast Beta				LTA Normal Power Fast Beta				LTA Low Power Beta				LTA Normal Power Beta			

- Bit 12-15: **LTA Low Power Beta Filter Value Fast**
  - 4-bit value
- Bit 8-11: **LTA Normal Power Beta Filter Value Fast**
  - 4-bit value
- Bit 4-7: **LTA Low Power Beta Filter Value**
  - 4-bit value
- Bit 0-3: **LTA Normal Power Beta Filter Value**
  - 4-bit value

Table A.8: Global Detection Settings

Global Detection Settings (0x34)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Number of Thresholds								Filter Halt Threshold							

- Bit 8-15: **Number of Thresholds**
  - The number of thresholds the max delta will be divided into
- Bit 0-7: **Filter Halt Threshold**
  - Ch0-3 Filter Halt threshold

Table A.9: Global Detection Hysteresis

Global Detection Hysteresis (0x35)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved												Hysteresis			

- Bit 0-3: **Trigger Level Hysteresis**
  - 4-bit hysteresis value.
  - $ReleaseThreshold = Threshold \times (1 - \frac{HysteresisValue}{15})$

Table A.10: Filter Halt and Key Press Timeout

Filter Halt and Key Press Timeout (0x3A)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Key Press Timeout								Filter Halt Timeout							

- Bit 8-15: **Key Press Timeout**
  - 8-bit value
  - $Timeout = value \times 500ms$
- Bit 0-7: **Filter Halt Timeout**
  - 8-bit value
  - $Timeout = value \times 500ms$

Table A.11: Key Scan Timeout

Key Scan Timeout (0x3B)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Key Scan Timeout								Key Scan I^{2}C Timeout							

- Bit 8-15: **Key Scan Timeout**
  - 8-bit value
  - Value in milliseconds
- Bit 0-7: **Key Scan I^{2}C Timeout**
  - 8-bit value
  - Value in milliseconds

Table A.12: Rx & Tx Selection

Rx & Tx Selection (0x3C,0x3D,0x3E,0x3F)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
CTX7	CTX6	CTX5	CTX4	CTX3	CTX2	CTX1	CTX0	Reserved				CRx3/ CRx7 Input	CRx2/ CRx6 Input	CRx1/ CRx5 Input	CRx0/ CRx4 Input

- Bit 15: **CTx7**
  - 0: CTx7 Tx pattern disabled
  - 1: CTx7 Tx pattern enabled
- Bit 14: **CTx6**
  - 0: CTx6 Tx pattern disabled
  - 1: CTx6 Tx pattern enabled
- Bit 13: **CTx5**
  - 0: CTx5 Tx pattern disabled
  - 1: CTx5 Tx pattern enabled
- Bit 12: **CTx4**
  - 0: CTx4 Tx pattern disabled
  - 1: CTx4 Tx pattern enabled
- Bit 11: **CTx3**
  - 0: CTx3 Tx pattern disabled
  - 1: CTx3 Tx pattern enabled
- Bit 10: **CTx2**
  - 0: CTx2 Tx pattern disabled
  - 1: CTx2 Tx pattern enabled
- Bit 9: **CTx1**
  - 0: CTx1 Tx pattern disabled
  - 1: CTx1 Tx pattern enabled
- Bit 8: **CTx0**
  - 0: CTx0 disabled
  - 1: CTx0 enabled
- Bit 3: **CRx3/CRx5 Input**
  - 0: CRx1/CRx5 input enabled
  - 1: CRx1/CRx5 input disabled
- Bit 2: **CRx1/CRx5 Input**
  - 0: CRx1/CRx5 input enabled
  - 1: CRx1/CRx5 input disabled
- Bit 1: **CRx1/CRx5 Input**

- 0: CRx1/CRx5 input enabled
- 1: CRx1/CRx5 input disabled

› Bit 0: **CRx0/CRx4 Input**
- 0: CRx0/CRx4 input enabled
- 1: CRx0/CRx4 input disabled

Table A.13: System Control

System Control (0x50)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved				I2C Pullup Resis-tors	Wait For Key Scan	Soft Reset	ACK Reset	Disable Key Scan	Zero Comp	Sync En-able	Reseed	Re-ATI	Power Mode Selection		

› Bit 11: **I²C pull-ups**
- 0: Internal I²C pull-ups are disabled
- 1: Internal I²C pull-ups are enabled

› Bit 9: **Wait for Key Scan**
- 0: Do not wait for key scan sequence before analogue cycles - Asynchronous analogue samples.
- 1: Wait for key scan sequence before analogue cycles - synchronous analogue samples.

› Bit 9: **Sofware reset**
- Set the bit to send a software reset.

› Bit 8: **Acknowledge Reset**
- Set bit to acknowledge an IC reset.

› Bit 7: **Disable Key Scanning**
- 0: Key scanning is enabled and I²C streaming is disabled.
- 1: Key scanning is disabled and I²C streaming is enabled.

› Bit 6: **Zero Compensation**
- 0: The ATI routine is allowed to add compensation to the channels.
- 1: The ATI routine is blocked from adding compensation to the channels.

› Bit 5: **Sync Enable**
- 0: Conversions occur at the set sampling rate.
- 1: Conversions occur with the falling edge of the sync line (GPIO 6).

› Bit 4: **Reseed**
- Set bit to set all channels' LTA equal to their filter counts.

› Bit 3: **Re-ATI**
- Set bit to trigger a re-ATI on all the channels.

› Bit 0-2: **Power Mode Selection**
- 001: Normal power mode
- 010: Low power mode
- 011: Ultra-Low power mode
- 100: Automatic power mode

Table A.14: Channel Control

Channel Control (0x51)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved				Ch3 Dis-abled	Ch2 Dis-abled	Ch1 Dis-abled	Ch0 Dis-abled	Ch3 Re-ATI	Ch2 Re-ATI	Ch1 Re-ATI	Ch0 Re-ATI	Ch3 Re-seed	Ch2 Re-seed	Ch1 Re-seed	Ch0 Re-seed

› Bit 11: **Ch3 Disabled**
- 0: Set Ch3 to enabled.
- 1: Set Ch3 to disabled.

› Bit 10: **Ch2 Disabled**
- 0: Set Ch2 to enabled.
- 1: Set Ch2 to disabled.

› Bit 9: **Ch1 Disabled**
- 0: Set Ch1 to enabled.
- 1: Set Ch1 to disabled.

› Bit 8: **Ch0 Disabled**
- 0: Set Ch0 to enabled.
- 1: Set Ch0 to disabled.

- Bit 7: **Ch3 Re-ATI**
  - Set bit to trigger a re-ATI on Ch3.
- Bit 6: **Ch2 Re-ATI**
  - Set bit to trigger a re-ATI on Ch2.
- Bit 5: **Ch1 Re-ATI**
  - Set bit to trigger a re-ATI on Ch1.
- Bit 4: **Ch0 Re-ATI**
  - Set bit to trigger a re-ATI on Ch0.
- Bit 3: **Ch3 Reseed**
  - Set bit to set Ch3's LTA equal to its filtered counts.
- Bit 2: **Ch2 Reseed**
  - Set bit to set Ch2's LTA equal to its filtered counts.
- Bit 1: **Ch1 Reseed**
  - Set bit to set Ch1's LTA equal to its filtered counts.
- Bit 0: **Ch0 Reseed**
  - Set bit to set Ch0's LTA equal to its filtered counts.

Table A.15: Conversion Frequency Setup

Conversion Frequency Setup (0x60)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Conversion Frequency Period								Conversion Frequency Fraction							

- Bit 7-14: **Conversion Frequency Fraction**
  - Set to 127
- Bit 8-15: **Conversion Frequency Period**
  - The calculation of the charge transfer frequency (f_{xfer}) is shown below. The relevant formula is determined by the value of the dead time enabled bit.
  - Dead time disabled: $f_{xfer} = \frac{f_clk}{2*period+2}$
  - Dead time enabled: $f_{xfer} = \frac{f_clk}{2*period+3}$
  - Range: 0 - 127
- Note: If the conversion frequency fraction is fixed at 127 and dead time is enabled, the following values of the conversion period will result in the corresponding charge transfer frequencies:
  - 0: 9.00MHz
  - 1: 4.50MHz
  - 2: 3.00MHz
  - 3: 2.25MHz
  - 4: 1.80MHz
  - 8: 1.00MHz
  - 17: 500kHz
- Note: The charge transfer frequency will be equal to the clock frequency (f_{osc}) if the fast mode bit is enabled in register 0x62.

Table A.16: Hardware Setup 0

Hardware Setup 0 (0x61)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Cs Size Select	Bias En-able	Pattern 1					Pattern 0					Reserved		Pad Control	

- Bit 15: **Cs Size Selection**
  - 0: 40pF
  - 1: 80pF
- Bit 14: **Bias Enable**
  - 0: Disable CRx8 bias voltage
  - 1: Enable CRx8 bias voltage
- Bit 9-13: **Pattern 1**
  - 0x0E: Normal Tx pattern
  - 0x0B: Inverse Tx pattern
- Bit 4-9: **Pattern 0**
  - 0x0E: Normal Tx pattern
  - 0x0B: Inverse Tx pattern

- Bit 0-1: **Pad control**
  - 0: Floating
  - 1: Floating
  - 2: Driven to VSS
  - 3: Driven to VREGA

Table A.17: Hardware setup 1

Hardware Setup 1 (0x62)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Pattern Selection								Mutual induc- tive diode	Coarse Div Mode		Dis- charge 0.5V	TG Fast Mode	TG Dead Time Enable	Init Len Selection	

- Bit 8-15: **Pattern Selection**
  - Assign either pattern 0 or pattern 1 to each of the CRx0-CRx7
  - 0: Use pattern 0
  - 1: Use pattern 1
- Bit 7: **Inductive Diode Enable**
  - 0: Inductive diode disabled
  - 1: Inductive diode enabled
- Bit 5-6: **Coarse Divider Mode**
  - 00: Normal coarse divider mode
  - 10: Inductive coarse divider mode
- Bit 4: **Discharge to 0.5V**
  - 0: Internal capacitors discharge to GND.
  - 1: Internal capacitors discharge to 0.5V.
- Bit 3: **TG Fast Mode**
  - 0: Inductive Tx signals are driven at the Rx Frequency.
  - 1: Inductive Tx signals are driven at the HFOSC Frequency.
- Bit 2: **TG Dead Time En**
  - Recommended to keep disabled.
  - 0: Disable conversion dead time.
  - 1: Enable conversion dead time.
- Bit 0-1: **Initial Length Selection**
  - Recommended to keep at 64 cycles
  - 0: 4 cycles
  - 1: 16 cycles
  - 2: 32 cycles
  - 3: 64 cycles

Table A.18: RF Blocking Settings

RF Blocking Settings (0x64)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Max Cycles blocked								RF blocking limit							

- Bit 8-15: **Max Cycles blocked**
  - 8-bit value indicating the maximum number of cycles to block possible RF noise for.
- Bit 0-7: **RF blocking limit**
  - 8-bit value indicating the delta step per sample which will be classified as RF noise.

Table A.19: Fine and Coarse Multipliers

Fine and Coarse Multipliers (0x70,0x72,0x74,0x76)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved		Fine Fractional Divider					Coarse Fractional Multiplier				Coarse Fractional Divider				

- Bit 9-13: **Fine Fractional Divider**
  - 5-bit value
- Bit 5-8: **Coarse Fractional Multiplier**
  - 4-bit value
- Bit 0-4: **Coarse Fractional Divider**
  - 5-bit value

Table A.20: ATI Compensation

ATI Compensation (0x71,0x73,0x75,0x77)															
Bit15	Bit14	Bit13	Bit12	Bit11	Bit10	Bit9	Bit8	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Compensation Divider					Res	Compensation Selection									

- Bit 11-15: **Compensation Divider**
  - 5-bit value
- Bit 0-9: **Compensation Selection**
  - 10-bit value

B Revision History

Release	Date	Changes
v0.1	6 June 2022	Initial release

C Known Issues

Contact Information

	USA	Asia	South Africa
Physical Address	11940 Jollyville Rd Suite 120-S Austin TX–78759 USA	Room501A, Block A T-Share International Centre Taoyuan Road Nanshan District Shenzhen Guangdong Province PRC	1 Bergsig Avenue Paarl 7646 South Africa South Africa
Postal Address	11940 Jollyville Rd Suite 120-S Austin TX–78759 USA	Room501A, Block A T-Share International Centre Taoyuan Road Nanshan District Shenzhen Guangdong Province PRC	PO Box 3534 Paarl 7620 South Africa South Africa
Tel	+1 512 538 1995	+86 755 8303 5294 ext 808	+27 21 863 0033
Email	info@azoteq.com	info@azoteq.com	info@azoteq.com

Visit www.azoteq.com
for a list of distributors and worldwide representation.