Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

IQS624 Datasheet

Combination sensor including: Hall-effect rotation sensing, along with dual-channel capacitive proximity/touch sensing, or single-channel inductive sensing.

The IQS624 ProxFusion® IC is a multifunctional capacitive and Hall-effect sensor designed for applications where any or all of the technologies may be required. The two Hall-effect sensors calculate the angle of a magnet rotating parallel with the sensor. The sensor is fully I²C compatible and on-chip calculations enable the IC to stream the current angle of the magnet without extra calculations.

Features

- **Hall effect angle sensor:**
  - On-chip Hall plates
  - 360° Absolute Output
  - 1° Resolution*, calculated on chip
  - Relative rotation angle.
  - Detect movement and the direction of movement.
  - Raw data: can be used to calculate degrees on external processor.
  - Wide operational range
  - No external components required
- **Partial auto calibration:**
  - Continuous auto-calibration, compensation for wear or small displacements of the sensor or magnet.
  - Flexible gain control
  - **A**utomatic **T**uning **I**mplementation (**ATI**) – Performance enhancement (10 bit).
- **Capacitive sensing**
  - Full auto-tuning with adjustable sensitivity
  - 2pF to 200pF external capacitive load capability

Inductive sensing
  - Only external sense coil required (PCB trace)
- **Multiple integrated UI**
  - Proximity / Touch
  - Proximity wake-up
  - Event mode
  - Wake Hall sensing on proximity
- Minimal external components
- Standard **I²C** interface
- Optional RDY indication for event mode operation
- **Low power consumption:**
- 240uA (100Hz response, Hall),
- 55uA (100Hz response, capacitive),
- 65uA (20Hz response, Hall)
- 15uA (20Hz response, capacitive)
- 5uA (5Hz response, capacitive)
- Supply Voltage: 2.0V to 3.6V**

*Optimal conditions
**5V solution available on demand.

DFN10
Representations only, not actual markings

Applications

- Anemometer
- Dial or Selector knob
- Mouse wheel
- Measuring wheel
- Digital angle gauge
- Speedometer for bicycle

Available Packages	
T_A	DFN(3x3)-10
-20°C to 85°C	IQS624-3yy1
-20°C to 85°C	IQS624-3yy2
-20°C to 85°C	IQS624-5yy**

Contents

List of abbreviations

PXS – ProxSense®

ATI – Automatic Tuning Implementation

LTA – Long term average

Thr – Threshold

UI – User interface

AC – Alternating current

DSP – Digital signal processing

RX – Receiving electrode

TX – Transmitting electrode

CS – Sampling capacitor

C – Capacitive

NP – Normal power

LP – Low power

ULP – Ultra low power

ACK – I^2C Acknowledge condition

NACK – I^2C Not Acknowledge condition

FG – Floating gate

1. Introduction

1.1 ProxFusion®

The ProxFusion® sensor series provide all the proven ProxSense® engine capabilities with additional sensors types. A combined sensor solution is available within a single platform.

1.2 Packaging and Pin-Out

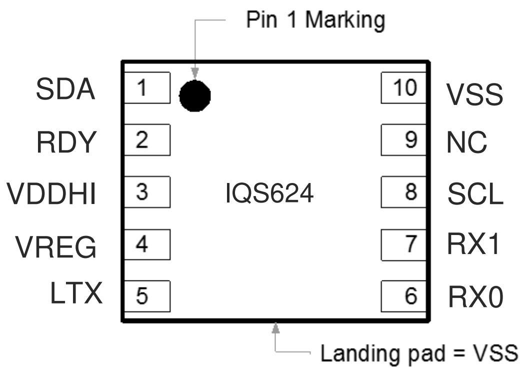


Figure 1-1 Pin out of IQS624 DFN (3X3)-10 package.

Table 1-1 IQS624 Pin-out

IQS624 Pin-out			
Pin	**Name**	**Type**	**Function**
1	SDA	Digital Input / Output	I^2C: SDA Output
2	RDY	Digital Output	I^2C: RDY Output
3	VDDHI	Supply Input	Supply Voltage Input
4	VREG	Regulator Output	Internal Regulator Pin
5	LTX	Analogue	Transmit Electrode 1
6	CRX0	Analogue	Sense Electrode 0
7	CRX1	Analogue	Sense Electrode 1
8	SCL	Digital Input / Output	I^2C: SCL Output
9	NC	Not connect	Not connect
10	VSS	Supply Input	Ground Reference

1.3 Reference schematic

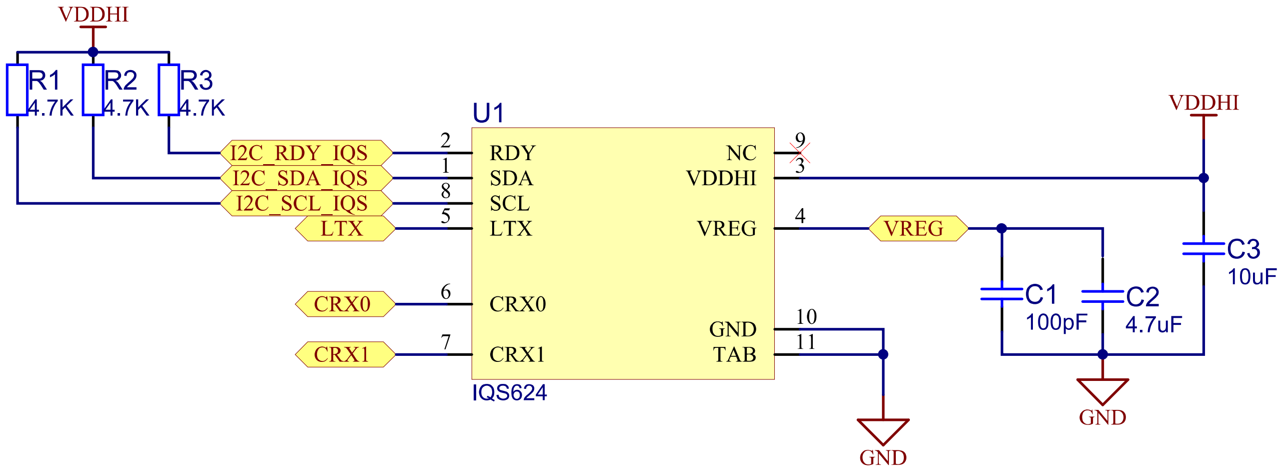


Figure 1-2 IQS624 reference schematic

1.4 Sensor channel combinations

The table below summarizes the IQS624's sensor and channel associations.

Table 1-2 Sensor - channel allocation

Sensor type	CH0	CH1	CH2	CH3	CH4	CH5
Discreet Self Capacitive	○	○				
Hall effect rotary UI			• 1st plate Positive	• 1st plate Negative	• 2nd plate Positive	• 2nd plate Negative
Mutual Inductive	○	○				

Key:

○ Optional implementation

• Fixed use for UI

1.5 ProxFusion® Sensitivity

The measurement circuitry uses a temperature stable internal sample capacitor (C_S) and internal regulated voltage (V_{REG}). Internal regulation provides for more accurate measurements over temperature variation. The size of the sample capacitor can be decreased to increase sensitivity on the capacitive channels of the IQS624.

$$Sensitivity \propto \frac{1}{C_s}$$

The Automatic Tuning Implementation (ATI) is a sophisticated technology implemented on the ProxFusion® series devices. It allows for optimal performance of the devices for a wide range of sense electrode capacitances, without modification or addition of external components. The ATI functionality ensures that sensor sensitivity is not affected by external influences such as temperate, parasitic capacitance and ground reference changes.

The ATI process adjusts three values (Coarse multiplier, Fine multiplier, Compensation) using two parameters (ATI base and ATI target) as inputs. A 10-bit compensation value ensures that an accurate target is reached. The base value influences the overall sensitivity of the channel and establishes a base count from where the ATI algorithm starts adding compensation. A rough estimation of sensitivity can be calculated as:

$$Sensitivity \propto \frac{Target}{Base}$$

As seen from this equation, the sensitivity can be increased by either increasing the Target value or decreasing the Base value. A lower base value will typically result in lower multipliers and more compensation would be required. It should, however, be noted that a higher sensitivity will yield a higher noise susceptibility. Refer to Appendix B and Appendix C for more information on Hall ATI.

2. Capacitive sensing

2.1 Introduction

Building on the previous successes from the ProxSense® range of capacitive sensors, the same fundamental sensor engine has been implemented in the ProxFusion® series.

2.2 Channel specifications

The IQS624 provides a maximum of 2 channels available to be configured for capacitive sensing. Each channel can be setup separately using the channel's associated settings registers.

Table 2-1 Capacitive sensing - channel allocation

Sensor type	CH0	CH1	CH2	CH3	CH4	CH5
Discreet Self Capacitive	○	○				

Key:

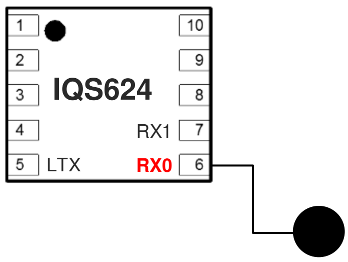
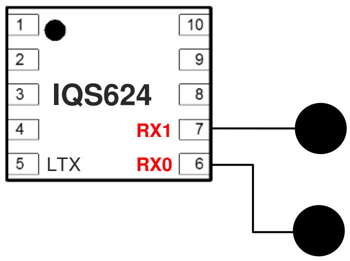
Optional implementation

- ○ Optional implementation
- • Fixed use for UI

2.3 Hardware configuration

In the table below are two options of configuring sensing (Rx) electrodes.

Table 2-2 Capacitive hardware description

	Self-capacitive configuration
1 button	
2 buttons	

2.4 Register configuration

2.4.1 Registers to configure for the capacitive sensing:

Table 2-3 Capacitive sensing settings registers

Address	Name	Description	Recommended setting
0x40, 0x41	Ch0/Ch1 ProxFusion Settings 0	Sensor mode and configuration of each channel.	Sensor mode should be set to capacitive mode An appropriate RX should be chosen and no TX
0x42	Ch0&Ch1 ProxFusion Settings 1	Global settings for the ProxFusion sensors	None
0x43, 0x44	Ch0/Ch1 ProxFusion Settings 2	ATI settings for ProxFusion sensors	ATI target should be more than ATI base to achieve an ATI
0x45	Ch0&Ch1 ProxFusion Settings 3	Additional Global settings for ProxFusion sensors	AC filter should be enabled
0x50, 0x52	Proximity threshold	Proximity Threshold for UI	Preferably more than touch threshold
0x51, 0x53	Touch threshold	Touch Threshold for UI	None

2.4.2 Proximity Thresholds

A proximity threshold for both channels can be selected for the application, to obtain the desired proximity trigger level. The proximity threshold is selectable between 1 (most sensitive) and 255 (least sensitive) counts. These threshold values (i.e. 1-255) are specified in Counts (CS) in the Ch0 Proximity threshold (0x50) and Ch1 Proximity threshold (0x51) registers for the discreet button UI.

2.4.3 Touch Thresholds

A touch threshold for each channel can be selected by the designer to obtain the desired touch sensitivity and is selectable between 1/256 (most sensitive) to 255/256 (least sensitive). The touch threshold is calculated as a fraction of the Long-Term Average (LTA) given by,

$$T_{THR} = {}^{x}/_{256} \times LTA$$

With lower target values (therefore lower LTA's) the touch threshold will be lower and vice versa.

Individual touch thresholds can be set for each channel, by writing to the Ch0 Touch threshold (0x51) and Ch1 Touch threshold (0x53) for the discreet button UI.

2.4.4 Example code:

Example code for an Arduino Uno can be downloaded at:

www.azoteq.com//images/stories/software/IQS62x_Demo.zip

2.5 Sensor data output and flags

The following register should be monitored by the master to detect capacitive sensor output.

a) The **Proximity/Touch UI Flags (0x12)** provide more detail regarding the outputs. A proximity and touch output bit for each channel 0 and 1 is provided in the Proximity/Touch UI Flags register.

Proximity/Touch UI Flags (0x12)								
Bit Number	7	6	**5**	**4**	3	2	**1**	**0**
Data Access	Read							
Name			**Chan 1 Touch out**	**Chan 0 touch out**			**Chan 1 proximity out**	**Chan 0 proximity out**

3. Inductive sensing

3.1 Introduction to inductive sensing

The IQS624 provides inductive sensing capabilities to detect the presence of metal/metal-type objects.

3.2 Channel specifications

The IQS624 requires 3 sensing lines for mutual inductive sensing.

A single inductance user interface is available.

a) Discreet proximity/touch UI (always enabled)

Table 3-1 Mutual inductive sensor – channel allocation

Mode	CH0	CH1	CH2	CH3	CH4	CH5
Mutual inductive	○	○				

Key:

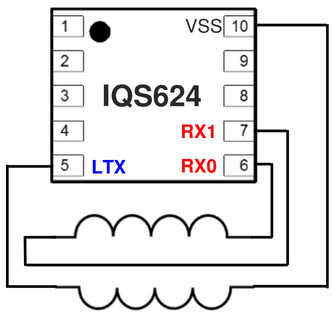
○ - Optional implementation

• - Fixed use for UI

3.3 Hardware configuration

Rudimentary hardware configurations (to be completed).

Table 3-2 Mutual inductive hardware description

<table>
<tr><th></th><th>Mutual inductive</th></tr>
<tr><td>Mutual inductance</td><td>
</td></tr>
</table>

3.4 Register configuration

Table 3-3 Inductive sensing settings registers.

Address	Name	Description	Recommended setting
0x40, 0x41	Ch0/Ch1 ProxFusion Settings 0	Sensor mode and configuration of each channel.	Sensor mode should be set to Inductive mode Choose one channel and deactivate the other channel Enable both RX for the activated channel
0x42	Ch0&Ch1 ProxFusion Settings 1	Global settings for the ProxFusion sensors	CS divider should be enabled
0x43, 0x44	Ch0/Ch1 ProxFusion Settings 2	ATI settings for ProxFusion sensors	ATI target should be more than ATI base to achieve an ATI
0x45	Ch0&Ch1 ProxFusion Settings 3	Additional Global settings for ProxFusion sensors	None
0x50, 0x52	Proximity threshold	Proximity Threshold for UI	Less than touch threshold
0x51, 0x53	Touch threshold	Touch Threshold for UI	None

3.4.1 Example code:

Example code for an Arduino Uno can be downloaded at:

www.azoteq.com//images/stories/software/IQS62x_Demo.zip

3.5 Sensor data output and flags

The following register should be monitored by the master to detect capacitive sensor output.

a) The **Proximity/Touch UI Flags (0x12)** provide more detail regarding the outputs. A proximity and touch output bit for each channel 0 and 1 is provided in the Proximity/Touch UI Flags register.

Proximity/Touch UI Flags (0x12)								
Bit Number	7	6	**5**	**4**	3	2	**1**	**0**
Data Access	Read							
Name			**Chan 1 Touch out**	**Chan 0 touch out**			**Chan 1 proximity out**	**Chan 0 proximity out**

4. Hall-effect sensing

4.1 Introduction to Hall-effect sensing

The IQS624 has two internal Hall-effect sensing plates (on die). No external sensing hardware is required for Hall-effect sensing.

The Hall-effect sensor measures the generated voltage difference across the plate, which can be modelled as a Wheatstone bridge. The voltage difference is converted to a current using an operational amplifier in order to be measured by the same ProxSense® sensor engine.

Advanced digital signal processing is performed to provide sensible output data.

- Calculates absolute position in degrees.
- Auto calibration attempts to linearize degrees output on the fly
- Differential Hall-Effect sensing:
    - Removes common mode disturbances

Refer to the Errata for correct setup of the IC.

4.2 Channel specifications

Channels 2 to 5 are dedicated to Hall-effect sensing. Channel 2 & 4 performs the positive direction measurements while channel 3 & 5 handle all measurements in the negative direction. Differential data is obtained from these four channels. This differential data is used as input data to calculate the output angle of the Hall-effect rotation UI. Channel 2 & 3 is used for the one plate and channel 4 & 5 for the second plate.

Table 4-1 Hall-effect sensor – channel allocation

Mode	CH0	CH1	CH2	CH3	CH4	CH5
Hall rotary UI			● 1st plate Positive	● 1st plate Negative	● 2nd plate Positive	● 2nd plate Negative

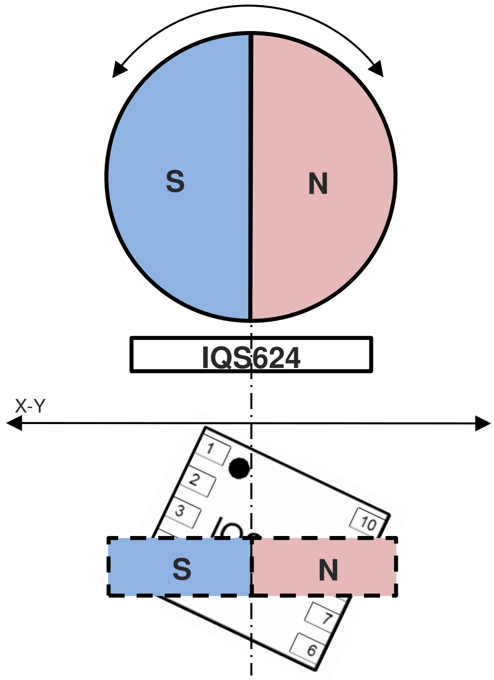
Key:

○ - Optional implementation

● - Fixed use for UI

4.3 Hardware configuration

Rudimentary hardware configurations. For more detail and alternative placement options, refer to Appendix A.

	Diametrically polarized magnet (rotational purposes)
Hall Rotation UI	

4.4 Register configuration

For more detail on the setup of the IQS624 refer to Appendix B.

Table 4-2 Hall sensing settings registers

Address	Name	Description	Recommended setting
0x70	Hall Rotation UI Settings	Hall wheel UI settings	Hall UI should be enabled for degree output; enable Auto Calibration
0x71	Hall sensor settings	Auto ATI and charge frequency settings	Auto ATI should be enabled for temperature drift compensation
0x72,0x73	Hall ATI Settings[1]	Hall channels ATI settings	ATI Target should be more than base
0x78	Hall ratio Settings	Invert Direction setting for Hall UI	None
0x79	Sin(phase) constant	Sin phase calibration value	Calculate this value using the GUI or the calculations in the Appendix A

[1] Refer to the errata and Appendix B

0x7A	Cos(phase) constant	Cos phase calibration value	Calculate this value using the GUI or the calculations in the Appendix A
0x7B[1]	Wheel Filter Beta	Degree filter value	Adjust filter value based on the amount of noise on the degree value
0x7C[1]	Wheel Wake Preload	Wheel wakeup settings	Use default values
0x7D[1]	Interval UI Divider	Divider for filtered degree values	Depending on the application (should be equal or greater than 3)
0x7E[1]	Wheel Offset	Wheel offset from zero position	Can be used for accurate intervals

4.4.1 Example code:

Example code for an Arduino Uno can be downloaded at:

www.azoteq.com//images/stories/software/IQS62x_Demo.zip

For ARM mbed resources refer to:

https://os.mbed.com/components/IQS624/

4.5 Sensor data output and flags

a) The **Hall UI Flags (0x14) register**. Bit7 is dedicated to indicating a movement of the magnet. Bit6 indicates the direction of the movement. Bit 1 is set when the movement counts are negative and bit 0 is set when the relative angle is negative. Bit 6 can be used to determine the magnet direction.

Hall UI Flags (0x14)								
Bit Number	**7**	**6**	5	4	3	2	1	0
Data Access	Read							
Name	**Wheel movement**	**Movement direction**						

b) The **Filtered Degree Output (0x17-0x16)** [1] A 16-bit value for the filtered degrees can be read from these registers. (0-359 degrees)

Filtered Degree Output[1] (0x17-0x16)																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data Access	Read															
Name	**Degrees High Byte**								**Degrees Low Byte**							

Bit definitions:

- 0-360: Filtered absolute degree position of magnet

c) The **Interval Number (0x18)** [1] An 8-bit value for the current interval number can be read from this register.

Interval Number[1] (0x18)								
Bit Number	7	6	5	4	3	2	1	0

[1] Only Available on IQS624-32

Data Access	Read/Write
Name	**Current Interval Number**

- Bit 7-0: Current interval number

d) The **Degree Output (0x81-0x80).** A 16-bit value for the degrees can be read from these registers. (0-360 degrees)

Degree Output (0x81-0x80)																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data Access	Read/Write															
Name	**Degrees High Byte**								**Degrees Low Byte**							

e) The **Relative Rotation Angle (0x8E).** The delta in degrees from the previous cycle to the current cycle can be read from this register. (0-180 degrees)

Relative Rotation Angle (0x8E)								
Bit Number	7	6	5	4	3	2	1	0
Data Access	Read/Write							
Name	**Relative degrees**							

4.6 IQS624-32 Interval UI

The IQS624-32 offers a new on-chip interval UI specifically designed for applications with discreet mechanical intervals or reduced resolution requirements.

4.6.1 Interval UI Features

- Adjustable interval size (3°-180°)
- The wheel can be zeroed at startup
- Adjustable wheel offset value
- An event is generated for changes to the Interval Number Register (0x18)

4.6.2 Interval UI Settings Registers

Hall Rotation UI Settings (0x70)								
Bit Number	7	6	5	4	3	2	1	0
Data Access	Read/Write							
Name	Hall Wheel UI disable	Interval UI disable[1]	**Zero Wheel[1]**	Hall Wheel Event disable[1]	**Interval Event Disable[1]**	Auto calibration	-	Wheel wakeup
UI Setting	0	0	**0**	**1**	**0**	1	-	0

Bit definitions:

- Bit 5: Zero Wheel angle[1]
  - 1: Zero angle, automatically cleared by firmware
- Bit 3: Interval UI Event disable[1]
  - 0: Event UI is enabled
  - 1: Event UI is disabled

[1] Only Available on IQS624-32

Interval UI Divider[1] (0x7D)								
Bit Number	7	6	5	4	3	2	1	0
Data Access	Read/Write							
Name	Interval size in degrees							
Default	3							

Bit definitions:

- Bit 7-0: Interval size in degrees (>= 3° and <=180°)
  - The **Interval Number (0x18)** [1] An 8-bit value for the current interval number can be read from this register.

Wheel Offset[1] (0x7E-0x7F)																
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data Access	Read															
Name	Wheel Offset High Byte								Wheel Offset Low Byte							
Default	0															

Bit definitions:

- 0-360: Wheel offset in degrees

4.6.3 Interval UI Output Register

Interval Number[1] (0x18)								
Bit Number	7	6	5	4	3	2	1	0
Data Access	Read/Write							
Name	Current Interval Number							

- Bit 7-0: Current interval number

4.7 IQS624-32 Example

Figure 4-1 illustrates an example wheel with 10 intervals. The IQS624-32 can be configured to generate an event on each interval. Configure the following registers for setup:

1. Disable Hall Wheel Event and Enable Interval Event in Hall Rotation Settings (0x70)
   - Write 0x14 to register 0x70
2. Set Interval UI Divider (0x7D) to 36D (360°/36° = 10 intervals)
   - Write 0x24 to register to 0x7D
3. If required, the wheel can be zeroed at startup in Hall Rotation UI Settings (0x70)
   - Set Bit5 in 0x70

[1] Only Available on IQS624-32

4. The interval register will increment every 36 degrees. If the wheel is zeroed using discrete mechanical intervals (Figure 4-1), half an interval should be added to the Wheel Zero Offset (0x7E-0x7F). With this offset, the interval register should increment when the wheel has moved half of the interval as shown by B in Figure 4-1.
5. Enable Event Mode in General System Settings (0xD0)
    - Set Bit5 in 0xD0
6. The interval can be read from the Interval Number (0x18) register.

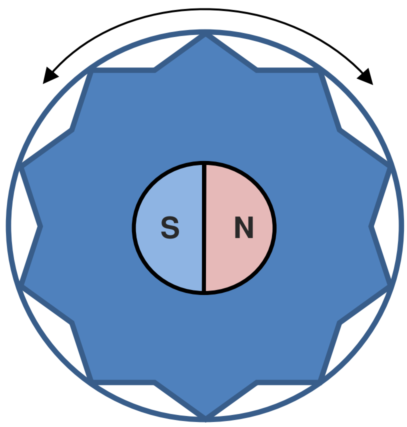


Figure 4-1: Discrete Mechanical Intervals

A 1-degree hysteresis is applied in the direction of rotation, resulting in two degrees of hysteresis at the interval change. This reduces the influence of jitter. In Figure 4-2, the interval will increase when point C is reached and will only decrease when the wheel moves back to point A. In this example A = 35°; B = 36°; C = 37°.

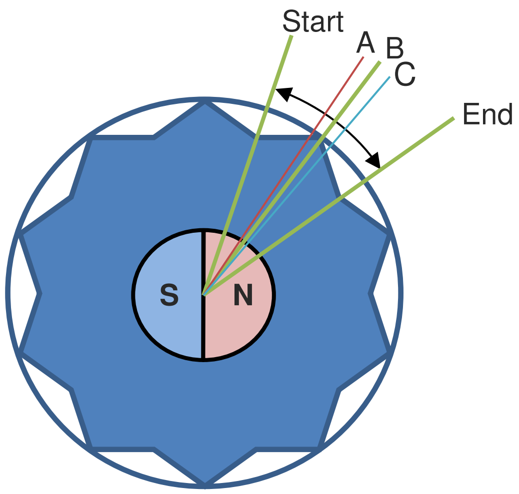


Figure 4-2: Interval UI Hysteresis

4.8 IQS624-32 Wheel Wake Preload (0x7C)

This register was added to improve performance. This register is compared to the Movement counter/timer (0x8F) register. The wheel will wake up if the counter value reaches the counter threshold value before the timer reaches 0. The timer in register 0x8F will count down from the value set in the **Wheel Wake Timer Preload.** The **Counter Threshold** is the amount of degrees the wheel has to move before an event is generated for movement. The wheel movement and movement direction bits in the Hall UI Flags (0x14) register also responds to this register.

The **Wheel Wake Timer Preload** can also be set to zero. In this mode the wheel will wake up if the wheel has moved the amount of degrees specified in the **Counter Threshold.** When the wheel is awake, 31 will be loaded in the **Wheel Wake Timer Preload**.

4.9 IQS624-32 Event Mode Options

The IQS624-32 provides three event mode options, these are:

1. **Touch on Wheel Wakeup enabled (Bit0, 0x70)**
   - The device wakes up from the low-power modes when there is a touch on Ch0.
   - In normal power mode events are only generated when there is a touch on Ch0. Ready events will be continuously generated as long as there is a touch on Ch0. If Touch on Wheel is enabled Interval and Hall Wheel Events are ignored.
2. **Interval Event (Bit3, 0x70)**
   - The device wakes up from low-power modes when the wheel increment or decrement the interval register.
   - Ready events are generated in normal power mode when the interval changes. If Interval Events and Hall Wheel Events (below) are enabled the IC will respond to Hall Wheel Events.
3. **Hall Wheel Event (Bit4, 0x70)**
   - The device wakes up based on the conditions described in Section 4.7. The wheel will wake up if the counter value reaches the counter threshold value before the timer reaches 0.
   - Events are generated on the same condition when the device is in low power mode. If Hall Wheel Events and Interval Events are enabled the IC will respond to Hall Wheel Events.

The device can also be set to **Stream in Normal Power (Bit5, 0xD9).** With this bit set the device will wake up from either of the selected modes above. Events will be generated based on the Normal Power Report Rate during Normal Power mode. The device will stop streaming when low-power mode is entered.

Table 4-3: Hall Rotation UI Settings

	Hall Rotation UI Settings (0x70)
Touch on Wheel Wakeup	0x1D
Interval Event	0x14
Hall Wheel Event	0x0C

If all of these modes are disabled, there will be no events generated for wheel movements.

5. Device clock, power management and mode operation

5.1 Device main oscillator

The IQS624 has a **16MHz** main oscillator (default enabled) to clock all system functionality.

An option exists to reduce the main oscillator to 8MHz. This will result in charge transfers to be slower by half of the default implementations.

To set this option:

- As a software setting – Set the General System Settings (0xD0): bit4 = 1, via an I^{2}C command.
- As a permanent setting – Set the OTP option in FG Bank 0: bit2 = 1, using Azoteq USBProg program.

The ProxFusion® channels charges at half of the main oscillator frequency. Therefore the frequency multiplier selected in Ch0&1 ProxFusion Settings 1 (0x42; bit 4-5) and Hall sensor settings (0x71; bit 4-5) is multiplied by half of the main oscillator frequency.

5.2 Device modes

The IQS624 supports the following modes of operation;

- **Normal mode** (Fixed report rate)
- **Low Power mode** (Reduced report rate, no UI execution)
- **Ultra-Low Power mode** (Only channel 0 is sensed for a prox)
- **Halt Mode** (Suspended/disabled)
  Note: Auto modes must be disabled to enter or exit halt mode.

The device will automatically switch between the different operating modes by default. However, this Auto mode feature may be disabled by setting the Disable Auto Modes bit (Power Mode Settings 0xD2; bit 5) to confine device operation to a specific power mode. The Power Mode bits (Power Mode Settings 0xD2; bit 3-4) can then be used to specify the desired mode of operation.

5.2.1 Normal mode

Normal mode is the fully active sensing mode to function at a fixed report rate specified in the Normal Mode report rate (0xD3) register. This 8-bit value is adjustable from 0ms – 255ms in intervals of 1ms.

5.2.2 Low power mode

Low power mode is a reduced sensing mode where all channels are sensed but no UI code are executed. The sample rate can be specified in the Low Power Mode report rate (0xD4) register. The 8-bit value is adjustable from 0ms – 255ms in intervals of 1ms. Reduced report rates also reduce the current consumed by the sensor.

5.2.3 Ultra-low power mode

Ultra-low power mode is a reduced sensing mode where only channel 0 is sensed and no other channels or UI code are executed. Set the Enable ULP Mode bit (Power Mode Settings 0xD2; bit 6) to enable use of the ultra-low power mode. The sample rate can be specified in the Low Power Mode report rate (0xD5) register. The 8-bit value is adjustable from 0ms – 4sec in intervals of 16ms.

When in Ultra-low power mode the IQS624 can be configured to update all channels at a specific rate defined in Power Mode Settings (0xD2) register. A flag will be set in the System flags (0x10;

bit 0) register when a normal power update is performed. Wake up will occur on proximity detection on channel 0. Ultra-low power mode will not function properly if channel 0 is not enabled.

5.2.4 Halt mode

Halt mode will suspend all sensing and will place the device in a dormant or sleep state. The device requires an I^2C command from a master to explicitly change the power mode out of the halt state before any sensor functionality can continue.

5.2.5 Mode time

The mode time is specified in the Auto Mode Timer (0xD6) register. The 8-bit value is adjustable from 0ms – 2 min in intervals of 500ms.

5.3 Streaming and event mode:

Streaming mode is the default. Event mode is enabled by setting bit 5 in the General System Settings (0xD0) register.

5.3.1 Streaming mode

The ready is triggered every cycle and per the report rate.

5.3.2 Event mode

The ready is triggered only when an event has occurred.

The events which trigger the ready can be configured to:

- Hall wheel movement (If the hall UI is enabled)
- Touch or proximity events on channel 0 or 1
- Interval Event[1]

Note: Both these events have built in hysteresis which filters out very slow changes.

[1] Only available on IQS624-32

5.4 Report rates

5.4.1 Normal Power Maximum Report rate

Note: Assuming normal mode report rate set to 0 (maximum speed) and Auto Power Modes turned off.

Hall UI State	Channels	Register Address	Bytes	Functionality[1]	Report Rate[2]
On	2 x Prox 4 x Hall	0x02 (PXS Flags) 0x80-0x81 (Degrees)	3	On-chip calculation of rotation angle and prox channels.	4.87 ms
On	4 x Hall	0x80-0x81 (Degrees)	2	On-chip calculation of rotation angle.	3.29 ms
Off	2 x Prox 4 x Hall	0x02 (PXS Flags) 0x24-0x2B (Counts)	9	Off-chip calculation of rotation angle and on-chip prox channels.	3.93 ms
Off	4 x Hall	0x24-0x2B (Counts)	8	Off-chip calculation of rotation angle.	2.94 ms
Off	1 x Hall 2 x Prox	0x24 (CH2 Counts) 0x02 (PXS Flags)	3	Off-chip RPM-calculation and 2 Prox channels on-chip	2.25 ms
Off	1 x Hall 1 x Prox	0x24 (CH2 Counts) 0x02 (PXS Flags)	3	Off-chip RPM-calculation and 1 Prox channels on-chip	1.63 ms
Off	1 x Hall	0x24 (CH2 Counts)	2	Off-chip RPM-calculation	0.82 ms

- Report rates are not necessarily an accurate indication of maximum observable rotation rate. On-chip calculations are only accurate at low rotation rates.

(1) Contact Azoteq for further information on functionality.
(2) These values were calculated by design and not by testing.

5.5 System reset

The IQS624 device monitor’s system resets and events.

a) Every device power-on and reset event will set the Show Reset bit in the System Flags (0x10; bit 7) register and the master should explicitly clear this bit by setting the Ack Reset bit in the General System Settings (0xD0; bit 6) register.

b) The system events will also be indicated with the Event bit in the System Flags (0x10; bit 1) register if any system event occur such as a reset. This event will continuously trigger until the reset has been acknowledged.

6. Communication

The **IQS624** device interfaces to a master controller via a 3-wire (SDA, SCL and RDY) serial interface bus that is I²C™ compatible with a maximum communication speed of 400 kHz. The communications interface of the IQS624 supports the following:

- Streaming data as well as event mode.
- The master may address the device at any time. If the IQS624 is not in a communication window, the device returns an ACK after which clock stretching is induced until a communication window is entered. Additional communication checks are included in the main loop in order to reduce the average clock stretching time.
- The provided interrupt line (RDY) is push-pull active low on IQS624-3001 and open-drain active low on IQS624-32. The RDY indicates a communication window.

6.1 Control Byte

The Control byte indicates the 7-bit device address (44H default) and the Read/Write indicator bit. The structure of the control byte is shown in Figure 6-1.

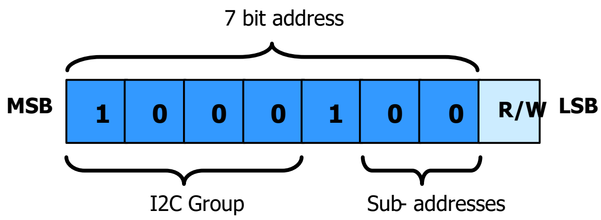


Figure 6-1 IQS624 Control Byte

The I²C device has a 7 bit Slave Address (default 0x44H) in the control byte. To confirm the address, the software compares the received address with the device address. Sub-address values can be set by OTP programming options.

6.2 I2C Read

To read from the device a *current address read* can be performed. This assumes that the address-command is already setup as desired.

Current Address Read

Start	Control Byte		Data n		Data n+1		Stop
S		ACK		ACK		NACK	S

Figure 6-2 Current Address Read

If the address-command must first be specified, then a *random read* must be performed. In this case a WRITE is initially performed to setup the address-command, and then a repeated start is used to initiate the READ section.

Random Read

Start	Control Byte		Address-command		Start	Control Byte		Data n		Stop
S	Adr + WRITE	ACK		ACK	S	Adr + READ	ACK		NACK	S

Figure 6-3 Random Read

6.3 I2C Write

To write settings to the device a *Data Write* is performed. Here the Address-Command is always required, followed by the relevant data bytes to write to the device.

DATA WRITE

Start	Control Byte		Address-Command		Data n		Data n+1		Stop
S	Adr + WRITE	ACK		ACK		ACK		ACK	S

Figure 6-4 I^2C Write

6.4 End of Communication Session / Window

Similar to other Azoteq I^2C devices, to end the I^2C communication session, a STOP command is given. When sending numerous read and write commands in one communication cycle, a repeated start command must be used to stack them together (since a STOP will jump out of the communication window, which is not desired).

The STOP will then end the communication, and the **IQS624** will return to process a new set of data. Once this is obtained, the communication window will again become available (RDY set LOW).

6.5 Stop-bit disable option(IQS624-32)

The IQS624-32 part offer:

- an additional I^2C settings register (0xD9) specifically added for stop-bit disable functionality,
- as well as a RDY timeout period register (0xD8) in order to set the required timeout period for termination of any communication windows (RDY = Low) if no I^2C activity is present on SDA and SCL pins.

Customers using an MCU with a binary serial-encoder peripheral which is not fully I^2C compatible (but provide some crude serial communication functions) can use this option to configure the IQS624-32 so that any auto generated stop command from the serial peripheral can be ignored by the IQS624-32 I^2C hardware. This will restrict the IQS624-32 from immediately exiting a communication window during event mode (reduced communication only for events) until all required communication has been completed and a stop command can correctly be transmitted. Please refer to the figures below for serial data transmission examples.

Please note:

1. Stop-bit disable and enable must be performed at the beginning and end of a communication window. The first and last I^2C register to be written to ensure no unwanted communication window termination.
2. Leaving the Stop-bit disabled will result in successful reading of registers but will not execute any commands written over I2C in a communication window being terminated after a RDY timeout and with no IQS recognised stop command.
3. The default RDY timeout period for IQS624-32 is purposefully long (10.24ms) for slow responding MCU hardware architectures. Please set this register according to your requirements/preference.
4. These options are only available on IQS624-32 parts and not for IQS624-3001.

Stop-bit Disable

Communication window open	Start	Control byte		Address-Command		Disable stop-bit		Ignored stop	Continue with reads / writes
RDY = ↓LOW	S	Addr + WRITE	ACK	0xDA	ACK	0x81	ACK	S	…

Figure 6-5 I²C Stop-bit Disable

Stop-bit Enable

Reads / Writes Finished	Start	Control byte		Address-Command		Enable stop-bit		Stop	Communication window closed
…	S	Addr + WRITE	ACK	0xDA	ACK	0x01	ACK	S	RDY = ↑HIGH

Figure 6-6 I²C Stop-bit Enable

6.6 Device address and sub-addresses

The default device address is **0x44 = DEFAULT_ADDR**.

Alternative sub-address options are definable in the following one-time programmable bits: **OTP Bank0 (bit3; 0; bit1; bit0) = SUB_ADDR_0** to **SUB_ADDR_7**

a) Default address: **0x44 = DEFAULT_ADDR OR SUB_ADDR_0**
b) Sub-address: **0x45 = DEFAULT_ADDR OR SUB_ADDR_1**
c) Sub-address: **0x46 = DEFAULT_ADDR OR SUB_ADDR_2**
d) Sub-address: **0x47 = DEFAULT_ADDR OR SUB_ADDR_3**
e) Sub-address: **0x4C = DEFAULT_ADDR OR SUB_ADDR_4**
f) Sub-address: **0x4D = DEFAULT_ADDR OR SUB_ADDR_5**
g) Sub-address: **0x4E = DEFAULT_ADDR OR SUB_ADDR_6**
h) Sub-address: **0x4F = DEFAULT_ADDR OR SUB_ADDR_7**

6.7 Additional OTP options

All one-time-programmable device options are located in FG bank 0.

Floating Gate Bank0								
Bit Number	7	6	5	4	3	2	1	0
Name	-	Comms ATI	-	-	Sub address 2	8MHz	Sub address 0-1	
Default	-	0	-	0	0	0	0	0

Bit definitions:

- Bit 6: Comms mode during ATI
  - 0: No streaming events are generated during ATI
  - 1: Comms continue as setup regardless of ATI state.
- Bit 2: Main Clock frequency selection
  - 0: Run FOSC at 16MHz
  - 1: Run FOSC at 8MHz
- Bit 0,1,3: I2C sub-address
  - I2C address = 0x44