



IQS7222D Arduino Example Code



Table of Contents

Introduction	1
Arduino Code Configuration	2
Example Code Flow Diagram	3
Sparkfun Board Library Installation	4
Serial Communication and Interface	7



Introduction

This Arduino example code demonstrates how to set up and use the IQS7222D Integrated Circuit (IC). The IQS7222D is a Versatile Trackpad/ touchscreen controller with proximity, touch, trackpad, and gesture outputs

This example code is specifically aimed at the IQS7222D Evaluation Kit (PCB number AZP1196A3).

This example code is intended for an Arduino Compatible board that uses 3.3 V logic, such as [Sparkfun's Pro Micro \(3.3V, 8 MHz\)](#). If a 5V logic Arduino board is used, a logic-level translator will be required between the Arduino-based board and the IQS7222D.



Arduino Code Configuration

In the example code folder, go to file: `src/IQS7222D/IQS7222D.h`. Change the value of the define to the number of the specific IQS7222D IC that the EV-KIT is populated with.

```
/* Device IC version select */
/* Select the IC version below by changing the value of the define (default = 0):
0: IQS7222D IC Version 1.2
1: IQS7222D IC Version 1.1
*/
#define IQS7222D_IC_VERSION 0
```

The behaviour and pin assignments of the Arduino code can be configured with the `#define` settings at the start of `iqs7222d-example-code.ino`.

Change the following pin assignments and parameters to suit your hardware:

```
/** Defines */
#define DEMO_IQS7222D_ADDR           0x48
#define DEMO_IQS7222D_POWER_PIN      4
#define DEMO_IQS7222D_RDY_PIN        7
```

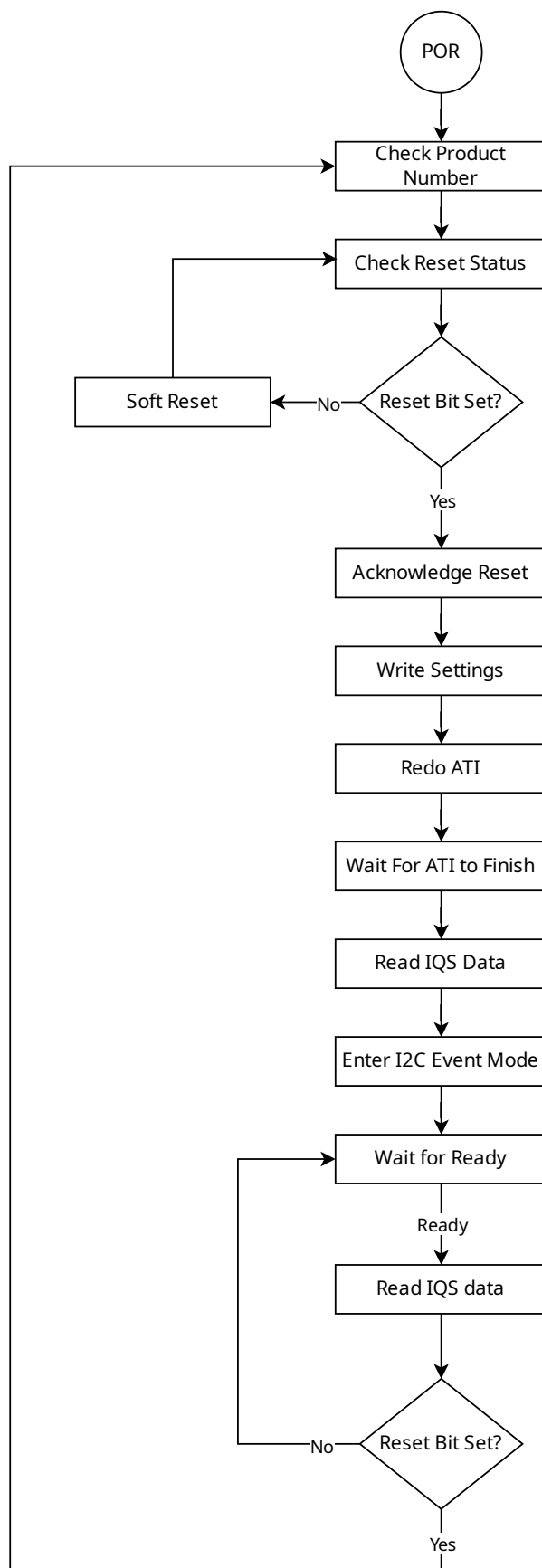
- `DEMO_IQS7222D_ADDR` is the IQS7222D I2C Slave address. For more information, refer to the datasheet and application notes found on the [IQS7222D Product Page](#).
- `DEMO_IQS7222D_POWER_PIN` can be used to power the IQS7222D directly from an Arduino GPIO. This parameter sets which pin to use. This is an optional setting and can be removed if the IQS7222D is powered from the VCC pin or an external power supply.
- `DEMO_IQS7222D_RDY_PIN` sets the pin assignment for the IQS7222D ready pin. This must support external interrupts. On the Sparkfun Pro Micro, pins 0, 1, 2, 3, and 7 support interrupts.



Please note that powering an IQS device directly from a GPIO is *generally* not recommended. However, the `DEMO_IQS7222D_POWER_PIN` in this example could be used as an enable input to a voltage regulator.



Example Code Flow Diagram





Sparkfun Board Library Installation

To use the Sparkfun Pro Micro, the Sparkfun Board Library must be installed in the Arduino IDE.

Add the Sparkfun Board Library by opening Preferences (**File > Preferences**), and paste the following URL into the "Additional Board Manager URLs" text box.

```
https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_Manager/package_sparkfun_index.json
```

Preferences

Settings Network

Sketchbook location: c:\Azoteq\Arduino **BROWSE**

☐ Show files inside Sketches

Editor font size: 14

Interface scale: ☒ Automatic 100 %

Theme: Light (Arduino)

Language: English (Reload required)

Show verbose output during ☐ compile ☐ upload

Compiler warnings: None

☐ Verify code after upload

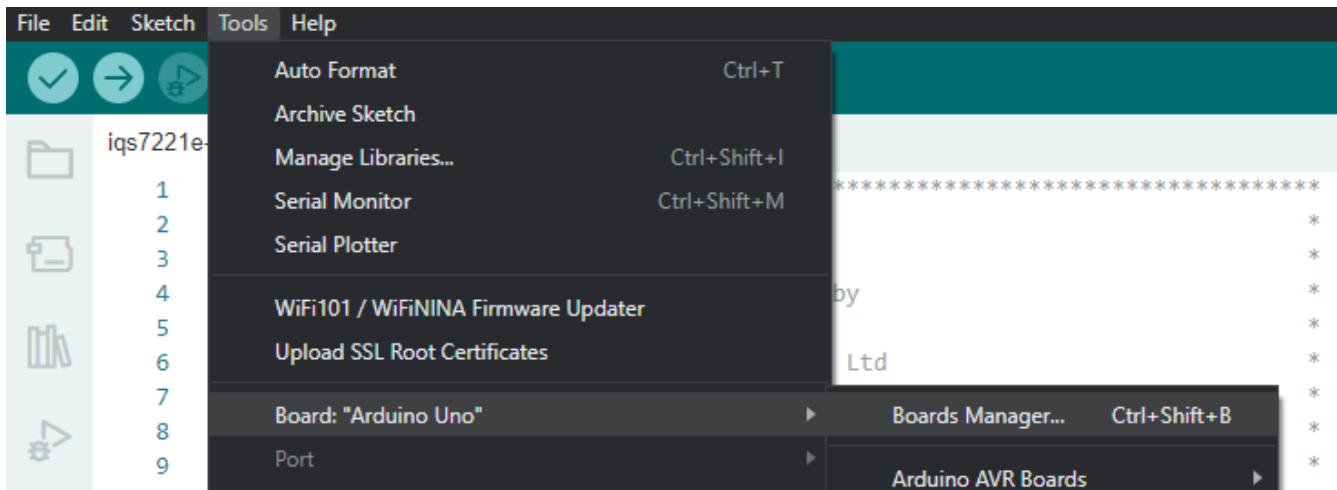
☒ Auto save

☐ Editor Quick Suggestions

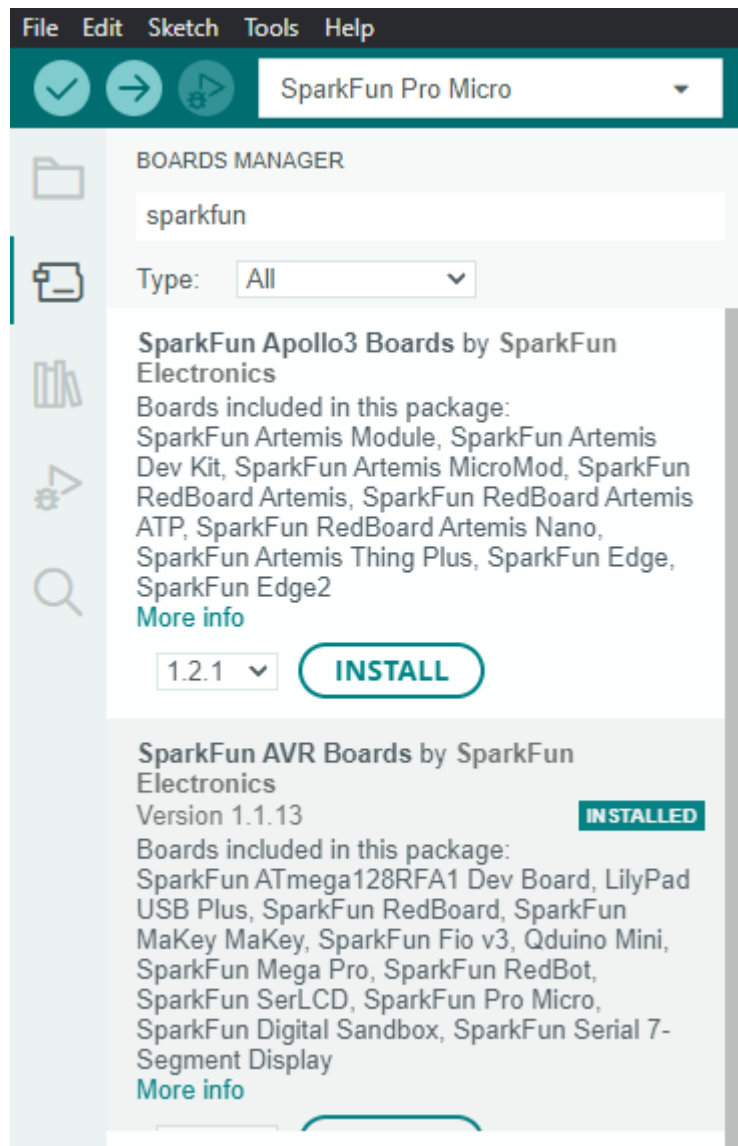
Additional boards manager URLs: https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_ **+**

CANCEL **OK**

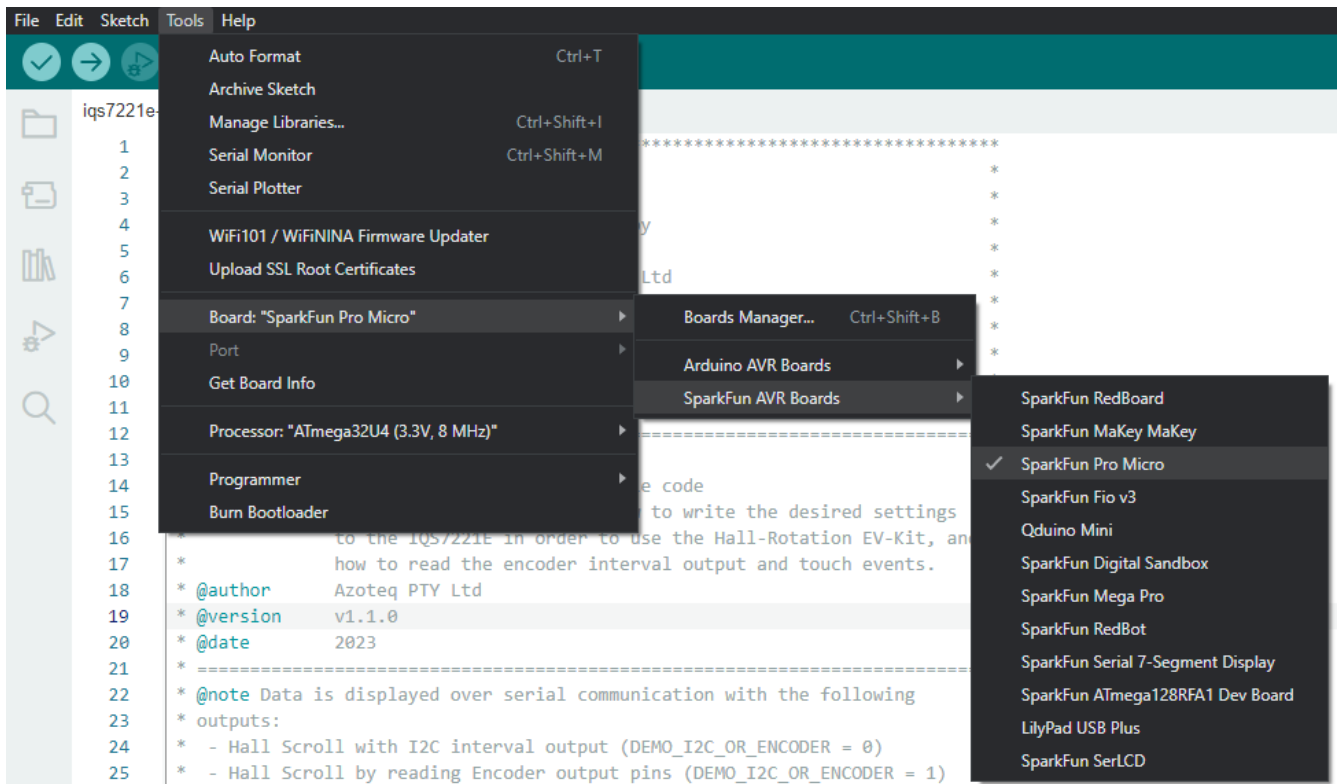
Click "OK". Then open the Board Manager under **Tools > Board > Boards Manager....**



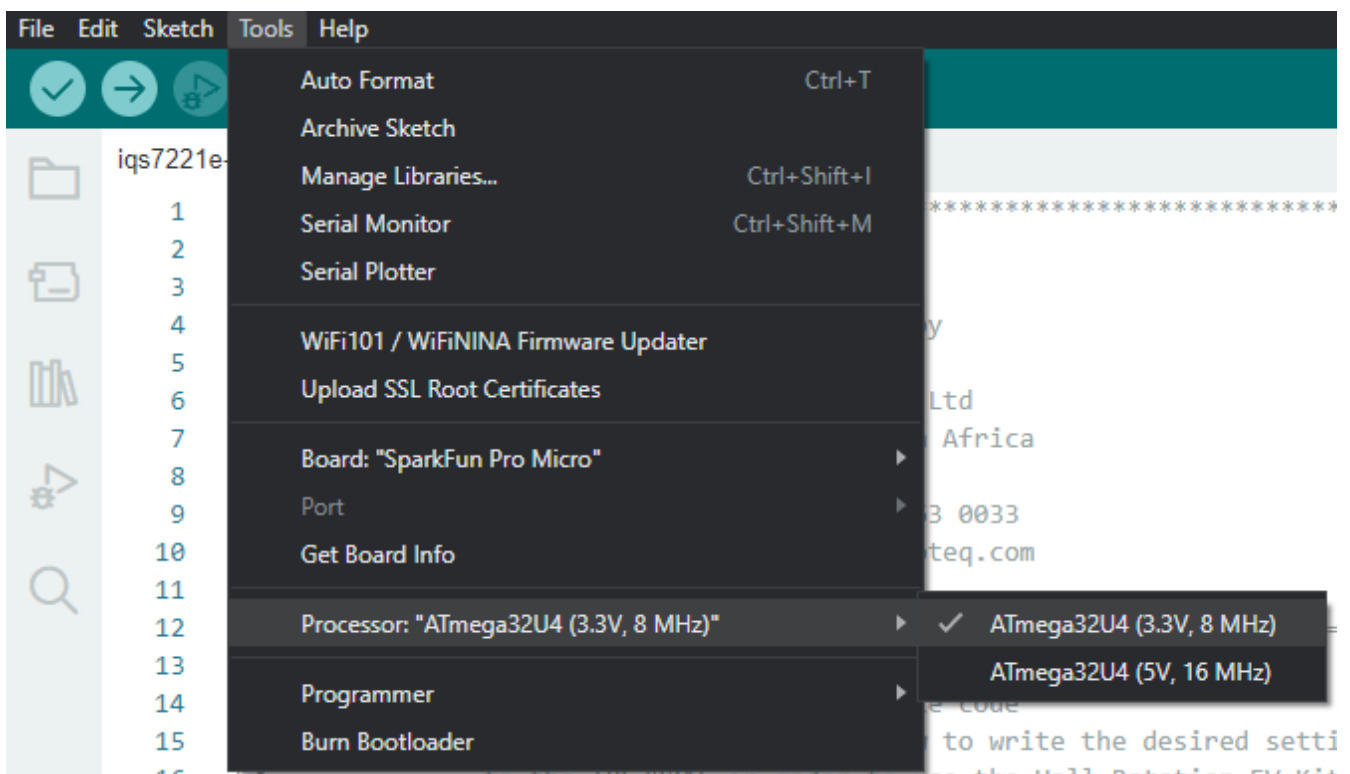
Search for "sparkfun", and install "SparkFun AVR Boards by SparkFun".



You can now select the "SparkFun Pro Micro" in the Board selection menu.



Also be sure to select the "3.3 V, 8 MHz" version under Tools > Processor.



Source: [Pro Micro Hookup Guide](#)



Serial Communication and Interface

The example code provides verbose serial feedback to aid in the demonstration of start-up and operational functions. It also has two built-in commands to demonstrate the IQS7222D's functionality. To use these built-in commands, the Arduino code simply sends an 'f' or 'r' over the serial interface.

1 - "f\n" - Force open a communication(RDY) window

2 - "r\n" - Request a Software Reset during runtime

It is important to take note of the newline ("\n") character that is needed to complete any serial request. It can be activated in the built-in Arduino IDE Serial monitor and is shown inside the blue rectangle in the figure below.

```
Start Serial communication
IQS7222D Ready
IQS7222D Initialization:
  IQS7222D_INIT_VERIFY_PRODUCT
    Product number is: 1046 v1.2
    IQS7222D Release UI Confirmed!
  IQS7222D_INIT_READ_RESET
    Reset event occurred.
  IQS7222D_INIT_UPDATE_SETTINGS
    1. Write Cycle Settings
    2. Write Button Settings
    3. Write Channel Setup Settings
    4. Write Filter Betas
    5. Write Hardware settings
    6. Write Deltas
    7. Write Gesture settings
    8. Write GPIO settings
    9. Write System Control Settings
  IQS7222D_INIT_ACK_RESET
  IQS7222D_INIT_ATI
  IQS7222D_INIT_WAIT_FOR_ATI
  DONE
  IQS7222D_INIT_READ_DATA
  IQS7222D_INIT_ACTIVATE_EVENT_MODE
  IQS7222D_INIT_DONE
IQS7222D Initialization complete!
```

☒ Autoscroll ☐ Show timestamp **Newline** 115200 baud Clear output