



IQS7211A 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 IQS7211A Integrated Circuit (IC). The IQS7211A is a Versatile Trackpad / touchscreen controller with proximity, touch, trackpad and gesture outputs. This example code accommodates both the rectangle- and flower trackpad Evaluation Kits.

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 IQS7211A.



Arduino Code Configuration

In the example code folder, go to file: `src/IQS7211A/IQS7211A.h`. Change the value of the define to the number of the specific IQS7211A EV-KIT the Arduino project needs to be compiled for.

```
/* Select the EV-Kit below by changing the value of the define (default = 0):  
0: Rectangle Trackpad EV-Kit (AZP1189A3).  
1: Flower Trackpad EV-Kit (AZP1191A3).  
*/  
#define IQS7211A_EV_KIT 0
```

The behaviour and pin assignments of the Arduino code can be configured with the `#define` settings at the start of `iqs7211a-example-code.ino`.

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

```
/** Defines */  
#define DEMO_IQS7211A_ADDR 0x56  
#define DEMO_IQS7211A_POWER_PIN 4  
#define DEMO_IQS7211A_RDY_PIN 7
```

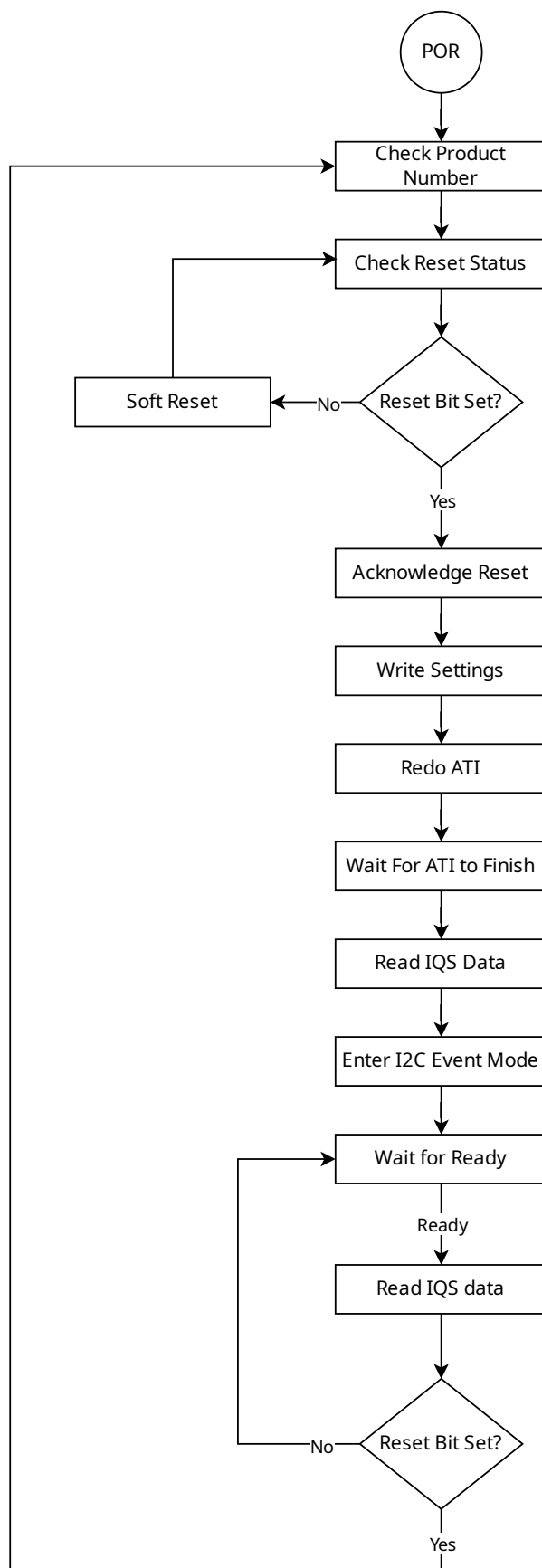
- `DEMO_IQS7211A_ADDR` is the IQS7211A I2C Slave address. For more information, refer to the datasheet and application notes found on the [IQS7211A Product Page](#).
- `DEMO_IQS7211A_POWER_PIN` can be used to power the IQS7211A directly from an Arduino GPIO. This parameter sets which pin to use. This is an optional setting and can be removed if the IQS7211A is powered from the VCC pin or an external power supply.
- `DEMO_IQS7211A_RDY_PIN` sets the pin assignment for the IQS7211A 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_IQS7211A_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

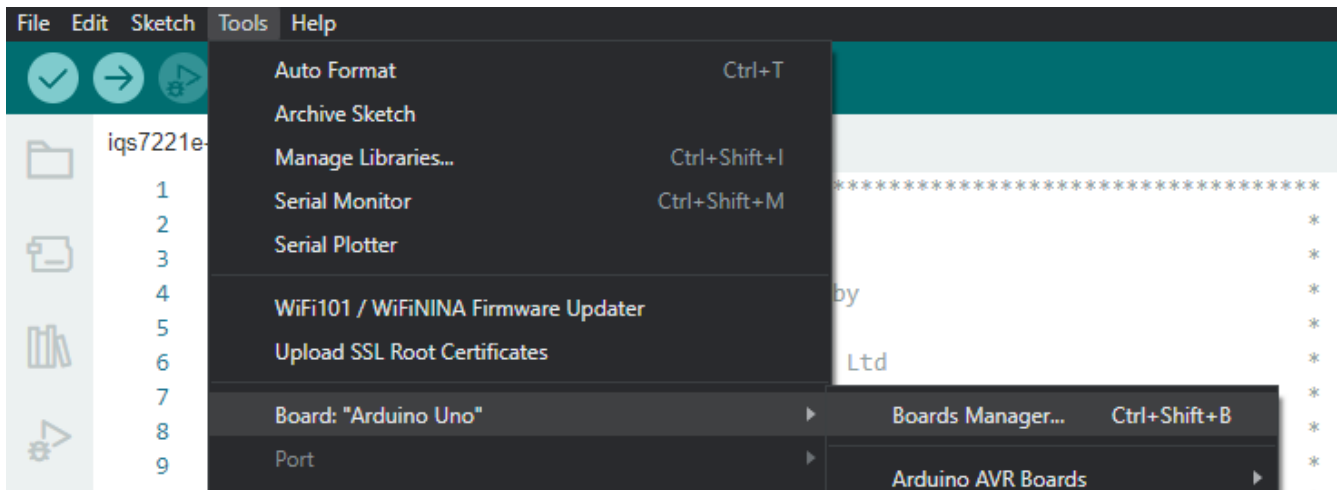
☒ Auto save

☐ Editor Quick Suggestions

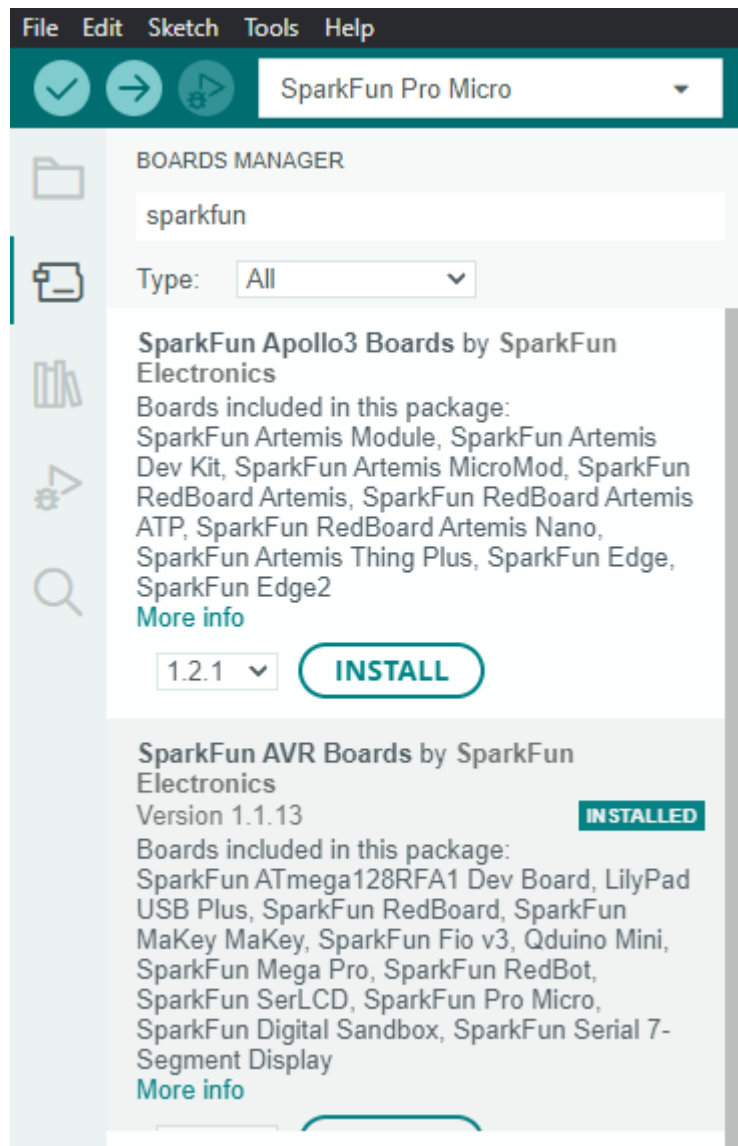
Additional boards manager URLs: https://raw.githubusercontent.com/sparkfun/Arduino_Boards/master/IDE_Board_ **+**

CANCEL **OK**

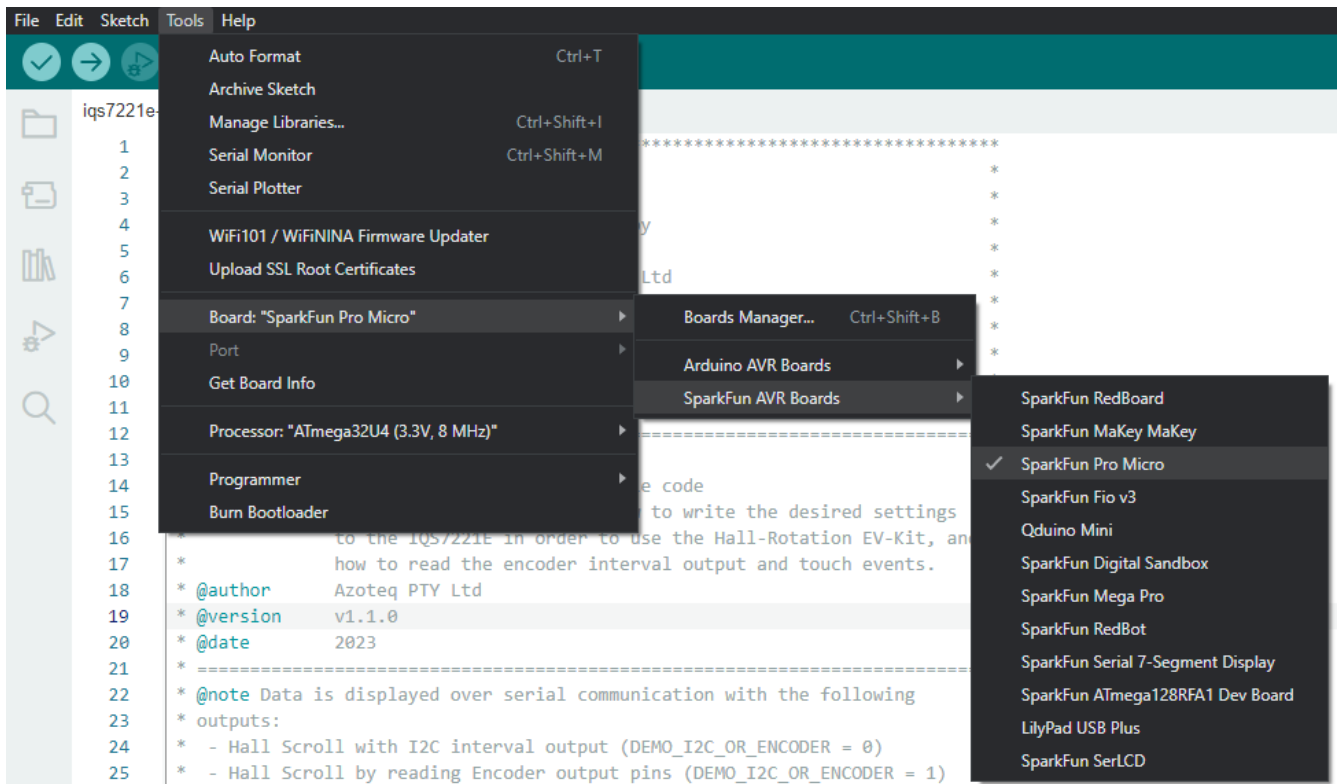
Click "OK". Then open the Board Manager under **Tools > Board > Boards Manager...**



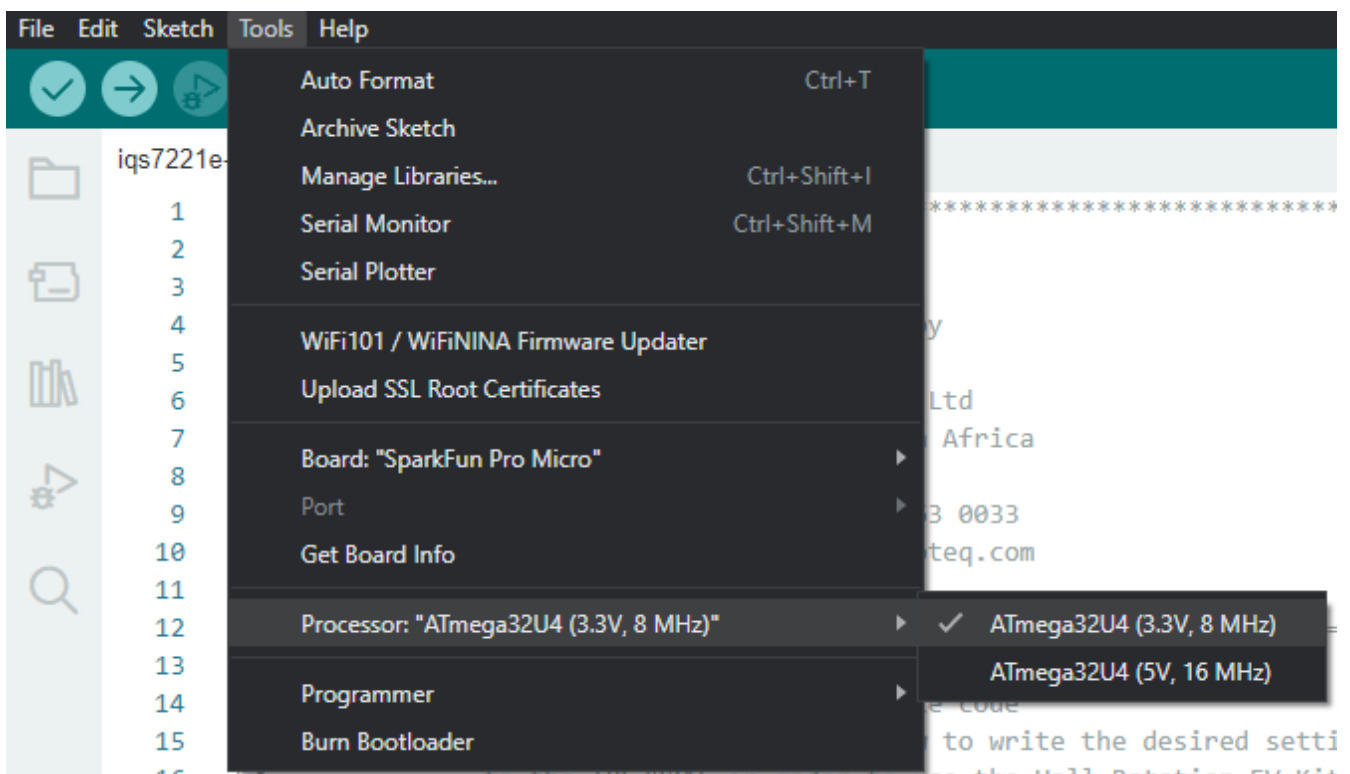
Search for "SparkFun", and install "SparkFun AVR Boards by SparkFun".



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 IQS7211A device's functionality. To use this built-in commands, the Arduino code simply sends an 'f' or an '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
IQS7211A Rectangle Touchpad EV-Kit Selected
IQS7211A Ready
IQS7211A Initialization:
  IQS7211A_INIT_VERIFY_PRODUCT
    Product number is: 763 v1.1
    IQS7211A Release UI Confirmed!
  IQS7211A_INIT_READ_RESET
    Reset event occurred.
  IQS7211A_INIT_UPDATE_SETTINGS
    1. Write ATI Settings
    2. Write ALP Compensation Settings
    3. Write Report rates and timings
    4. Write System control settings
    5. Write Hardware settings
    5. Write Filter Betas
    6. Write Gesture Settings
    7. Write TP Settings
    8. Write Cycle 0 - 9 Settings
    9. Write Cycle 10 - 17 Settings
  IQS7211A_INIT_ACK_RESET
  IQS7211A_INIT_ATI
  IQS7211A_INIT_WAIT_FOR_ATI
    DONE
  IQS7211A_INIT_READ_DATA
  IQS7211A_INIT_ACTIVATE_EVENT_MODE
  IQS7211A_INIT_DONE
IQS7211A Initialization complete!
```

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