RM3100 Arduino Quick Guide

Table of Contents

[1 RM3100 Arduino Overview 2](#_Toc88636886)

[2 Hardware connection and set up 2](#_Toc88636887)

[3 Install Arduino IDE software 6](#_Toc88636888)

[4 Run RM3100 Arduino code 8](#_Toc88636889)

|  |  |  |
| --- | --- | --- |
| *Version #* | *Changes* | *Date* |
| Draft | Initial Version Draft | 11/24/2021 |

# RM3100 Arduino Overview

PNI developed a demo program using the Arduino IDE on Win PC for customers to get started using the RM3100 with an Arduino. RM3100 supports I2C and SPI, it’s up to customer to decide which interface to use. Connection between RM3100 and Arduino can be either I2C or SPI, Arduino as master and RM3100 as slave for both.

To run this demo program, following is the minimum requirement.

**Hardware**

RM3100 Evaluation Board or Break Board

Any Arduino (In this document, we used the [NUCLEO-L152RE](https://www.st.com/en/evaluation-tools/nucleo-l152re.html#overview) which supports Arduino)

Win7 or later PC

**Software**

[Arduino IDE](https://www.arduino.cc/en/software)

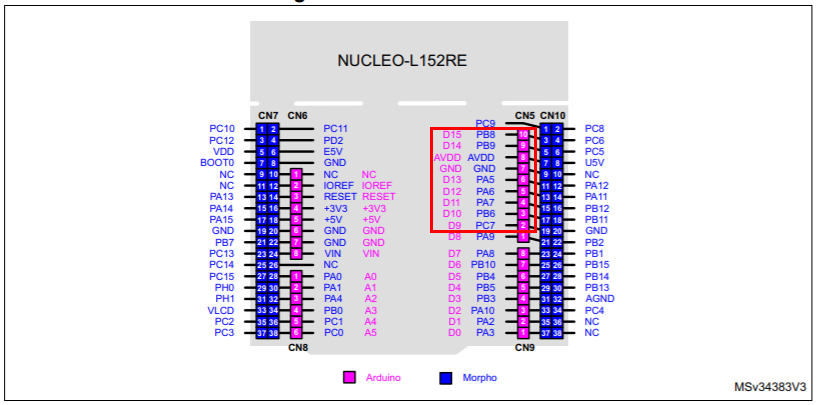
# Hardware connection and set up

1. RM3100 Test Boards Pin Assignments (Pin numbers run counter clockwise from top left)

|  |  |  |
| --- | --- | --- |
| Pin# | Pin Name | Description |
| 1 | SCK /  SCL | SPI interface (SCK) – Serial clock input  I2C interface (SCL) – Serial clock line |
| 2 | SO /  SA1 | SPI interface (SO) – Master Input, Slave Output  I2C interface – Bit 1 of slave address |
| 3 | SI / SDA | SPI interface (SI) – Master Output, Slave Input Serial Data  I2C interface (SDA) – Serial Data Line |
| 4 | SSN /  SA0 | SPI interface – Active low to select port  I2C interface – Bit 0 of slave address |
| 5 | DRDY | Status line |
| 7 | AVSS | Ground pin for analog section of ASIC |
| 10 | I2CEN | I2C enable pin (HIGH = I2C, LOW = SPI) |
| 12 | DVDD | Supply voltage for digital section of ASIC. |
| 13 | AVDD | Supply voltage for analog section of ASIC |
| 14 | DVSS | Ground pin for digital section of ASIC |
| 6, 8, 9, 11 | NC | Do not connect |

1. NUCLEO-L152RE SPI and I2C Pin Assignments
   1. Note: If you are using a regular Arduino board (Nano, Uno, Mega, etc.) you will have to find the correct corresponding pins from your specific board’s data sheet as they will be different from this table below.

|  |  |  |  |
| --- | --- | --- | --- |
| Pin# | | Pin Name | Description |
| D15 | SCL | | I2C interface (SCL) – Serial clock line |
| D14 | SDA | | I2C interface (SDA) – Serial Data Line |
| AVDD | AVDD/+3V3 | | Power (+3.3 V) |
| GND | GND | | Ground |
| D13 | SCK | | SPI Serial Clock |
| D12 | MISO | | SPI Master In Slave Out |
| D11 | MOSI | | SPI Master Out Slave In |
| D10 | CS | | Slave Select control line that allows slaves to be turned on and off via hardware control |
| D9 | GPIO | | General Purpose Input pin used for status line |



Red box above highlights the pins used for I2C and SPI communications on NUCLEO-L152RE

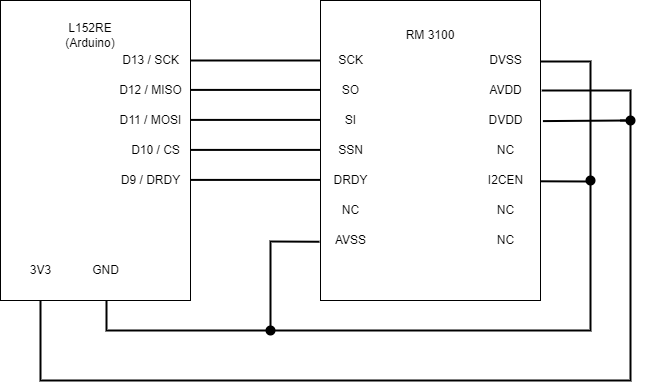
1. I2C Interface. Connect RM3100 I2C pins to Arduino I2C pins, like in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **RM3100 I2C Pins** | | | **Arduino I2C** |
| **Pin#** | **Pin Name** | **Description** | **Pin#** |
| 1 | SCL | I2C interface (SCL) – Serial clock line | D15 SCL |
| 2 | SA1 | I2C interface – Bit 1 of slave address (LOW = 0) | GND |
| 3 | SDA | I2C interface (SDA) – Serial Data Line | D14 SDA |
| 4 | SA0 | I2C interface – Bit 0 of slave address (LOW = 0) | GND |
| 5 | DRDY | Status line (Optional) | D9 (GPIO input) |
| 7 | AVSS | Ground pin for analog section of ASIC | GND |
| 10 | I2CEN | I2C enable pin (HIGH = I2C) | AVDD (+3V3) |
| 12 | DVDD | Supply voltage for digital section of ASIC | AVDD (+3V3) |
| 13 | AVDD | Supply voltage for analog section of ASIC | AVDD (+3V3) |
| 14 | DVSS | Ground pin for digital section of ASIC | GND |
| 6, 8, 9, 11 | NC | Do not connect |  |

Note: If you are using I2C, you may have to enable the internal pullup resistors in the code for your specific board. The sample code successfully enables pullup resistors for the NUCLEO-L152RE board specifically and hasn’t been tested with other boards such as the Arduino Uno or Arduino Mega. You can also use external 4.7k Ohm pull up resistors on the SCL and SDA lines instead of the internal resistors.

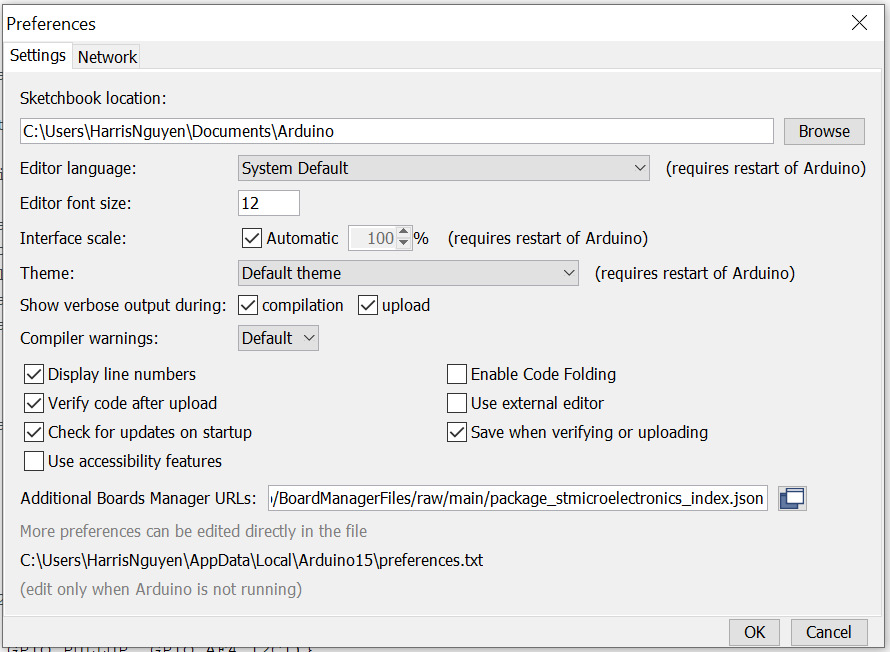
1. If SPI interface is desired, please connect RM3100 SPI pins to Arduino SPI pins, like in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **RM3100 SPI Pins** | | | **Arduino SPI** |
| **Pin#** | **Pin Name** | **Description** | **Pin#** |
| 1 | SCK | SPI interface (SCK) – Serial clock input | D13 SCK |
| 2 | SO | SPI interface (SO) – Master Input, Slave Output | D12 MISO |
| 3 | SI | SPI interface (SI) – Master Output, Slave Input | D11 MOSI |
| 4 | SSN | SPI interface – Active low to select port | D10 SS |
| 5 | DRDY | Status line (Optional) | D9 (GPIO input) |
| 7 | AVSS | Ground pin for analog section of ASIC | GND |
| 10 | I2CEN | I2C enable pin (HIGH = I2C, LOW=SPI) | GND |
| 12 | DVDD | Supply voltage for digital section of ASIC | AVDD (+3V3) |
| 13 | AVDD | Supply voltage for analog section of ASIC | AVDD (+3V3) |
| 14 | DVSS | Ground pin for digital section of ASIC | GND |
| 6, 8, 9, 11 | NC | Do not connect |  |



1. Plug in Arduino USB cable to the USB port on your PC

# Install Arduino IDE software

1. To use the demo code, you will first need to have the Arduino IDE installed. If you don’t have it installed, here is a link to [version 1.8.16 of Arduino IDE](https://downloads.arduino.cc/arduino-1.8.16-windows.exe?_gl=1*1b559pi*_ga*NDQ2NDU3NTUuMTYzNDkyMzQyNA..*_ga_NEXN8H46L5*MTYzNzYwNjkzMy40MC4xLjE2Mzc2MDg4OTEuMA). You can also check [the IDE downloads page](https://www.arduino.cc/en/software) to see if there are any newer updates but the demo code was written on version 1.8.16, so it may be incompatible with newer versions. Open the \*.exe and install the program using any settings you want.
2. If you are using a STM 32 board (like the NUCLEO-L152RE in this guide) then you must install the STM 32 board package first. If you are using a regular Arduino board (UNO, Mega, etc.) you can skip to step III.
   1. Open the Arduino IDE program and in the toolbars press File > Preferences
   2. In “Additional Boards Manager URLs:” copy and paste: <https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json> and press OK to close the Preferences window
   3. In the toolbars press Tools > Board: > Boards Manager
   4. Graphical user interface, text, application, email

      Description automatically generatedSearch “stm 32” and install the latest version of “STM32 MCU based boards” then close the Boards Manager page
   5. Now select the corresponding NUCLEO board group in Tools > Board: and then the specific board in Tools > Board part number:
   6. Text

      Description automatically generatedFinally while the Arduino is connected to the PC, press Tools > Port and pick the USB port the Arduino is connected to (this should be something similar to “COM 6” or “COM 3”). Your “Tools” dropdown should look something like this:
3. If you are using a standard Arduino (Nano, Uno, Mega, etc.), open the Arduino IDE program and in the tool bar press Tools > Boards and choose your specific Arduino board. Also, while the Arduino is connected to the PC, press Tools > Port and pick the USB port the Arduino is connected to (this should be something like “COM 6” or “COM 3”). Your “Tools” dropdown should look something like this:

Graphical user interface, application

Description automatically generated

# Run RM3100 Arduino code

1. Graphical user interface, text, application, email

   Description automatically generatedIn the Arduino IDE, File > Open “**RM3100\_Arduino\_I2C.ino**” or “**RM3100\_Arduino\_SPI.ino**” file depending on which communication setup you have decided on. The code should open and look like this:
2. The red box above highlights the options you can change.
   1. The option initialCC is the cycle count you want the RM 3100 to measure in.
      1. The internal clock count of RM3100 MagI2C ASIC establishes the number of sensor oscillation cycles. The number of oscillation cycle is “Cycle Count”. Cycle Count as default value is 200 or 0xC8 in hex.
      2. Increasing the cycle count value increases measurement gain and resolution. Lowering the cycle count value reduces acquisition time,

which increases maximum achievable sample rate or, with a fixed sample rate, decreases power consumption.

* + 1. The minimum value is ‘0’ and the maximum is 65,536. However, quantization issues generally dictate working above a cycle count value of ~30, while noise limits the useful upper range to ~400 cycle counts.
  1. The option “singleMode” is used to decide which measurement mode to use: single measurement mode or continuous measurement mode
     1. Setting singleMode to “0” will enable continuous measurement mode
     2. Setting singleMode to “1” will enable single measurement mode
  2. The option “useDRDYPin” is used to decide which method to use to wait for data to be ready
     1. Setting “useDRDYPin” to “0” will make the Arduino read from the internal status register to see if data is ready
     2. Setting “useDRDYPin” to “1” will make the Arduino read from the Pin\_DRDY (pin D9) to see if data is ready
        1. If you use the DRDY pin then you must have pin D9 connected to RM 3100 pin 5 (DRDY)

1. Once you have the desired options selected, press upload button (Outlined in red in the top left of the image below) to verify and upload the code on to the Arduino
   1. Graphical user interface, text, application, email

      Description automatically generatedYou can also use “Ctrl + U” or press Sketch > Upload to upload
2. After the code is uploaded to the Arduino, press the orange highlighted button in the top right of the image above or press Tools > Serial Monitor or “Ctrl + Shift + M” to open the serial monitor. This is where the output of the RM 3100 will be printed

Text

Description automatically generated