# **RAK13009 Quick Start Guide**

#### **Prerequisite**

## What Do You Need?

Before going through each and every step on using the RAK13009 WisBlock QWIIC, make sure to prepare the necessary items listed below:

#### Hardware

- RAK13009 WisBlock QWIIC
- Your choice of WisBlock Base 
   I
- Your choice of WisBlock Core 
  ☐
- USB Cable
- QWIIC-based 16x2 LCD
- RAK19008 WisBlock IO Extension Cable(optional) 1
- Li-Ion/LiPo battery (optional)
- Solar charger (optional)

#### Software

- Download and install the Arduino  $\mathsf{IDE}\, \square$  .
- To add the RAKwireless Core boards on your Arduino Boards Manager, install the RAKwireless Arduino BSP ☐
   .

# **Product Configuration**

## **Hardware Setup**

The RAK13009 is a QWIIC module, which is part of the RAKWireless WisBlock Interface series. This module has two connectors: one for the WisBlock sensor and then for the standard QWIIC interface. For more information about RAK13009, refer to the Datasheet.

The RAK13009 WisBlock QWIIC can be mounted on any Sensor slot of the WisBlock Base board, as shown in **Figure 1**. Also, always secure the connection of the WisBlock module by using compatible screws.



Figure 1: RAK13009 Connection to WisBlock Base

#### Assembling and Disassembling of WisBlock Modules

#### Assembling

As shown in **Figure 2**, the location for sensor slots is properly marked by silkscreen. Follow carefully the procedure defined in RAK5005-O module assembly/disassembly instructions I<sup>™</sup> to attach a WisBlock module. Once attached, carefully fix the module with three pieces of M1.2 x 3 mm screws.



Figure 2: RAK13009 assembly to WisBlock Base

#### Disassembling

The procedure in disassembling any type of WisBlock modules is the same.

1. First, remove the screws.



Figure 3: Removing screws from the WisBlock module

2. Once the screws are removed, check the silkscreen of the module to find the correct location where force can be applied.



Figure 4: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 5**, to detach the module from the base board.

![](_page_2_Picture_1.jpeg)

Figure 5: Applying even forces on the proper location of a WisBlock module

#### **NOTE**

If you will connect other modules to the remaining WisBlock Base slots, check on the WisBlock Pin Mapper 2 tool for possible conflicts.

After all this setup, you can now connect the battery (optional) and USB cable to start programming your WisBlock Core.

#### **WARNING**

- Batteries can cause harm if not handled properly.
- Only 3.7-4.2 V Rechargeable LiPo batteries are supported. It is highly recommended not to use other types of batteries with the system unless you know what you are doing.
- If a non-rechargeable battery is used, it has to be unplugged first before connecting the USB cable to the USB port of the board to configure the device. Not doing so might damage the battery or cause a fire.
- Only 5 V solar panels are supported. Do not use 12 V solar panels. It will destroy the charging unit and eventually other electronic parts.
- Make sure the battery wires match the polarity on the RAK5005-O board. Not all batteries have the same wiring.

#### **Software Configuration and Example**

![](_page_3_Picture_1.jpeg)

Figure 6: RAK13009 and 16x2 LCD using QWIIC interface

In this example, you will be able to see the I2C address of the device you connected using the QWIIC interface.

1. You need to select first the WisBlock Core you have, as shown in Figure 7 to Figure 9.

File Edit Sketch Tools Help       Auto Format     Ctrl+T       Archive Sketch     Ctrl+T	©- ▼
Auto Format Ctrl+T Archive Sketch	<b>₽</b>
sketch_aug2     Fix Encoding & Reload       1     void s       2     // p       Serial Plotter     Ctrl+Shift+I	^
4 } WiFi101 / WiFiNINA Firmware Updater	
5       void       1       Board: "WisBlock Core RAK4631 Board"       Boards Manager         6       void       1       Bootloader: "0.3.2 SoftDevice s140 6.1.1"       Arduino AVR Boards         7       // p       Debug: "Level 0 (Release)"       Arduino AVR Boards       Arduino SAMD (32-bits ARM Cortex-M0+) Boards >         8       Port: "COM4"       ESP8266 Boards (3.0.2)       >         9       Get Board Info       ESP8266 Boards (3.10-dev) >       >         Programmer       RAKwireless ESP32 Modules       >         Burn Bootloader       RAKwireless nRF Modules       WisBlock Core RAK4631 Board	
	~

Figure 7: Selecting RAK4631 as WisBlock Core

🥯 sketch_aug22c	Arduino 1.8.15				-	đ	$\times$
File Edit Sketch To	pols Help						
	Auto Format	Ctrl+T					. <u>O</u>
	Archive Sketch						
sketch_aug2:	Fix Encoding & Reload						•
1 void s	Manage Libraries	Ctrl+Shift+I					^
2 // n	Serial Monitor	Ctrl+Shift+M					
2 // P	Serial Plotter	Ctrl+Shift+L					
3	MEETOT / MEETNING Engineer Lindeter						
4 }	WIFITOT / WIFININA FIRMware Opdater						
5	Board: "WisCore RAK11200 Board"	>	Boards Manager				
6 void 1	Upload Speed: "921600"	>	Arduino AVR Boards	>			
7 // p	Flash Frequency: "80MHz"	>	Arduino SAMD (32-bits ARM Cortex-M0+) Boards	>			
8	Flash Mode: "QIO"	>	ESP8266 Boards (3.0.2)	>			
9}	Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)	" >	ESP8266 Boards (3.1.0-dev)	>			
	Core Debug Level: "None"	>	RAKwireless ESP32 Modules	WisCore RAK11200 Board			
	Port: "COM4"	>	RAKwireless nRF Modules	>			
	Get Board Info						
	Programmer	>					
	Burn Bootloader						
							•
1			WisCore R&K11200 Board D	efault 4MB with sniffs (1 2MB APP/1 5MB	SPIEES) OIO 80MHz 921600	None on (	сома —
			Wiscold NART 200 Doard, Dr		or in they, early, continue, by 1000	, Hone on e	

Figure 8: Selecting RAK11200 as WisBlock Core

∞ sketch_oct29a   A	rduino 1.8.15					_	٥	×
File Edit Sketch Too	bls Help							
	Auto Format	Ctrl+T						<mark>بو</mark> ر
	Archive Sketch							_
sketch_oct29	Fix Encoding & Reload							
1 void s	Manage Libraries	Ctrl+Shift+I						^
2 // n	Serial Monitor	Ctrl+Shift+M	once:					
3	Serial Plotter	Ctrl+Shift+L	onee.					
4 }	WiFi101 / WiFiNINA Firmware Update	r						
5	Board: "WisBlock RAK11300"	>	Boards Manager					
6 void 1	Port: "COM10 (WisBlock RAK11300)"	>	Arduino AVR Boards					
7 // p	Get Board Info		Arduino SAMD (32-bits ARM Cortex-M0+) Boards					
8	Programmer	,	ESP8266 Boards (3.0.2)					
9 }	Burn Bootloader		ESP8266 Boards (3.1.0-dev)					
	bum bootioader		RAKwireless ESP32 Modules					
			RAKwireless nRF Modules					
			Rakwireless Raspberry Modules	WisBlock RAK113	300			
								~
1						WisBlock RAK	11300 on CO	M10

Figure 9: Selecting RAK11300 as WisBlock Core

2. On the Arduino IDE, go to Sketch > Include Library > Manage Libraries. The Library Manager should open, then install the LiquidCrystal I2C 🖸 library, as shown in Figure 10.

## **BAK**<sup>®</sup> Documentation Center

🕺 Library Manager	>
Type All V Topic All V liquid crystal i2d	
graphic displays (u8glib), More info	
	Version 2.3.5 V Install
LiquidCrystal I2C	
by Marco Schwartz A library for I2C LCD displays. The library allows to control I2C displays w	with functions extremely similar to LiquidCrystal library. THIS LIBRARY
MIGHT NOT BE COMPATIBLE WITH EXISTING SKETCHES.	
More info	
LiquidCrystal_AIP31068	
by Andriy Golovnya	
A library for AIP31068 I2C/SPI LCD displays. The library allows to con- to LiquidCrystal library. THIS LIBRARY MIGHT NOT BE COMPATIBLE WITH EXI	trol AIP31068 based I2C/SPI displays with functions extremely similar STING SKETCHES.
More info	
LiquidCrystal 12C Hangul	
by Junwha Hong	
A library for printing Hangul on I2C LCD displays. The library allows to c library. This Library allows to print hangul on LCDs.	ontrol I2C displays with functions extremely similar to LiquidCrystal
More info	
	Close
	Close

Figure 10: Installing the LiquidCrystal I2C library

3. After successful installation of the library, you can now copy the following sample code into your Arduino IDE:

```
#include LiquidCrystal_I2C.h
#include Wire.h
//the third parameter is how many columns are on your screen
LiquidCrystal_I2C lcd(0x27, 16, 2); //change the 0x27 based on the result from the I2C scanner co
void setup() {
 lcd.init(); //initialize lcd screen
 lcd.backlight(); // turn on the backlight
ļ
void loop() {
 start_display(); // star
 delay(1000); //wait for a second
 lcd.clear(); // clear the LCD content
 delay(1000); //wait for a second
}
void start_display(){
 lcd.setCursor(0,0); // tell the screen to write on the top row
 lcd.print("QWIIC"); // tell the screen to write "QWIIC" on the top row
 lcd.setCursor(0,1); // tell the screen to write on the bottom row
 lcd.print("EXAMPLE"); // tell the screen to write "EXAMPLE" on the bottom row
```

4. Select the right Serial Port and upload the code, as shown in Figure 11 and Figure 12.

TEST_CODE	Arduino 1.8.15					_		×
File Edit Sketch To	ools Help							
	Auto Format	Ctrl+T						<b>1</b> 0
	Archive Sketch							
RAK13002_L	Fix Encoding & Reload							
1 #inclu	Manage Libraries	Ctrl+Shift+I						^
2	Serial Monitor	Ctrl+Shift+M						
3#inclu	Serial Plotter	Ctrl+Shift+L						
4	WiFi101 / WiFiNINA Firmware Updater							
5//init	Board: "WisBlock Core RAK4631 Board"	>						
6//the	Bootloader: "0.3.2 SoftDevice s140 6.1.1"	>	Б					
7//the	Debug: "Level 0 (Release)"	>	are on yo	our screen				
8//the	Port: "COM8"	>	Serial ports	your screen				
9 Liquid	Get Board Info		COM3					
10	Programmer	X	COM8					
11 void s	Burn Bootloader							
12 lcd.i	<pre>nit(); //initialize lo</pre>	cd screen	1					
13 lcd.ba	acklight(); // turn on t	he backl	iaht					$\sim$
Done Saving.								
6				WisBlock Core RAK4	631 Board, 0.3.2 SoftDevice s140 6.	.1.1, Level 0 (Re	elease) on C	ОМВ

Figure 11: Selecting the correct Serial Port

TEST CODE   Arduino 1.8.15		×
File Edit Sketch Tools Help		
V Deload		P
RAK13002_LCD §		
1 #include LiquidCrystal_I2C.h		^
2		
3 #include Wire.h		
4		
5//initialize the liquid crystal library		
6//the first parameter is the I2C address		
7 //the second parameter is how many rows are on your screen		
8 //the third parameter is how many columns are on your screen		
<pre>9 LiquidCrystal_I2C lcd(0x27, 16, 2);</pre>		
10		
11 void setup() {		
12 lcd.init(); //initialize lcd screen		
13 lcd.backlight(); // turn on the backlight		~
Done Saving.		
5 WisBlock Core RAK4631 Board, 0.3.2 SoftDevice s140 6.1.1, Level 0	(Release) on	сомв

![](_page_6_Figure_4.jpeg)

#### **NOTE:**

RAK11200 requires the BOOT0 pin to be configured properly before uploading. If not done properly, uploading the source code to RAK11200 will fail. Check the full details on the RAK11200 Quick Start Guide.

5. When you have successfully uploaded the sample code, you should see this on your LCD QWIIC display:

![](_page_7_Picture_1.jpeg)

Figure 13: QWIIC EXAMPLE displayed on 16x2 LCD

6. If you are not seeing the same output, check the device's I2C address by using this code:

#### **BAK**<sup>°</sup> Documentation Center

```
* Follow the connection of LCD with I2C Backpack to RAK13002.
#include <Wire.h> //include Wire.h library
void setup()
 Wire.begin(); // Wire communication begin
 Serial.begin(9600); // The baudrate of Serial monitor is set in 9600
 while (!Serial); // Waiting for Serial Monitor
 Serial.println("\nI2C Scanner");
void loop()
  byte error, address; //variable for error and I2C address
  int nDevices;
  Serial.println("Scanning...");
  nDevices = 0;
  for (address = 1; address < 127; address++ )</pre>
   Wire.beginTransmission(address);
   error = Wire.endTransmission();
   if (error == 0)
     Serial.print("I2C device found at address 0x");
     if (address < 16)
       Serial.print("0");
      Serial.print(address, HEX);
     Serial.println(" !");
     nDevices++;
   else if (error == 4)
      Serial.print("Unknown error at address 0x");
     if (address < 16)
       Serial.print("0");
     Serial.println(address, HEX);
  if (nDevices == 0)
   Serial.println("No I2C devices found\n");
   Serial.println("done\n");
 delay(5000); // wait 5 seconds for the next I2C scan
```

7. Your device's I2C address should be displayed on the Serial Monitor, as shown in Figure 14.

© COM8				-		]	$\times$
						S	end
22:51:22.058 -> I2C device found at address 0x	k27 !						1
22:51:22.058 -> done							
22:51:22.058 ->							
22:51:27.073 -> Scanning							
22:51:27.073 -> I2C device found at address 0x	k27 !						
22:51:27.073 -> done							
22:51:27.073 ->							
22:51:32.088 -> Scanning							
22:51:32.088 -> I2C device found at address 0x	k27 !						
22:51:32.088 -> done							
22:51:32.088 ->							
22:51:37.101 -> Scanning							
22:51:37.101 -> I2C device found at address 0x	k27 !						
22:51:37.101 -> done							
22:51:37.101 ->							
							~
Autoscroll Show timestamp		Both NL	& CR ~	9600 baud	~	Clear ou	tput

Figure 14: I2C address of your 16x2 LCD

Last Updated: 7/29/2022, 10:17:19 PM

# **RAK13009 WisBlock QWIIC Module Datasheet**

### **Overview**

### Description

The RAK13009 is a QWIIC module, which is part of the RAKWireless WisBlock Interface series. This module has two connectors: one for the WisBlock sensor and then for the standard QWIIC interface. By using this module you can plug any QWIIC interface module into the WisBlock sensor and use it just like any other WisBlock sensor.

### Features

- WisBlock Sensor module
- Standard QWIIC interface
- Module Size: 10 mm x 10 mm

# Specifications

### Overview Mounting

The RAK13009 WisBlock QWIIC can be mounted to the IO slot of the WisBlock Base board. **Figure 1** shows the mounting mechanism of the RAK13009 on a WisBlock Base board, such as the RAK5005-O

![](_page_10_Picture_12.jpeg)

Figure 1: RAK13009 WisBlock QWIIC mounting

### Hardware

The hardware specification is categorized into four parts. It shows the pinouts and their corresponding functions and diagrams. It also covers the electrical and mechanical parameters that include the tabular data of the functionalities and standard values of the RAK13009 WisBlock QWIIC Module.

#### **Pin Definition**

The RAK13009 WisBlock QWIIC comprises a standard WisBlock sensor connector. The WisBlock sensor connector allows the RAK13009 module to be mounted to a WisBlock baseboard, such as RAK5005-O. The pin order of the connector and the pinout definition is shown in **Figure 2**.

#### 📝 NOTE

• I2C related pin, INT pin, 3V3\_S, and GND are connected to WisBlock sensor connector

## **BAK**<sup>®</sup> Documentation Center

![](_page_11_Figure_1.jpeg)

Figure 2: RAK13009 WisBlock QWIIC pinout

## Electrical Characteristics QWIIC Interface Wiring Label

![](_page_11_Figure_4.jpeg)

#### Figure 3: QWIIC cable

All the QWIIC cables follow the color scheme and arrangement, as shown in the table below.

Wire Color	Label
Black	GND
Red	3.3 V
Blue	SDA
Yellow	SCL

#### 

The recommended max current on a QWIIC cable is 226 mA.

# **Mechanical Characteristics**

#### **Board Dimensions**

Figure 4 shows the dimensions and the mechanical drawing of the RAK13009 module.

# **BAK**<sup>°</sup> Documentation Center

![](_page_12_Figure_1.jpeg)

Figure 4: RAK13009 dimensions

![](_page_12_Figure_3.jpeg)

Figure 5: QWIIC interface dimensions (in mm)

### WisConnector PCB Layout

![](_page_13_Figure_1.jpeg)

![](_page_13_Figure_2.jpeg)

![](_page_13_Figure_3.jpeg)

Figure 7: RAK13009 schematic diagram