RAK13600 Quick Start Guide

Prerequisite

What Do You Need?

Before going through each and every step on using the RAK13600 WisBlock module, make sure to prepare the necessary items listed below:

Hardware

- RAK13600 Wisblock Interface Extension Board with Coil Antenna
- Your choice of WisBlock Base
 I
- Your choice of WisBlock Core
 ☐
- USB Cable
- Li-Ion/LiPo battery (optional)
- Solar charger (optional)

Software

- Download and install the ArduinoIDE \square .
- To add the RAKwireless Core boards on your Arduino Boards Manager, install the RAKwireless Arduino BSP

Product Configuration

Hardware Setup

The RAK13600 module is designed as wireless module that allows you to scan NFC and RFID tags and devices. It includes an antenna coil that transmits and receives RF signals from the object being scanned. Without the antenna coil shown in **Figure 1**, you will not be able to scan NFC/RFID devices. The antenna coil has adhesive transfer tape on it. You can remove the 3M patch film so you stick the antenna coil on the ideal location in your enclosure. The enclosure must be plastic and not metal, since metal enclosures introduce attenuation on RF signals.



Figure 1: NFC Coil Antenna

The RAK13600 module can be mounted on the IO slot of any WisBlock Base board, as shown in **Figure 2**. Also, always secure the connection of the WisBlock module by using compatible screws. For more information about RAK13600, refer to the Datasheet.



Figure 2: RAK13600 on WisBlock Base with WisBlock Core

Assembling and Disassembling of WisBlock Modules Assembling

As shown in **Figure 3**, the location for the IO slot is properly marked by silkscreen. Follow carefully the procedure defined in WisBlock Base module assembly/disassembly instructions \square to attach a WisBlock module. Once attached, carefully fix the module with three M1.2 x 3 mm screws compatible with the module.



Figure 3: RAK13600 connection to WisBlock Base Board

Disassembling

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

1. First, remove the screws.



Figure 4: 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 5: Detaching silkscreen on the WisBlock module

3. Apply force to the module at the position of the connector, as shown in **Figure 6**, to detach the module from the baseboard.



Figure 6: 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 C tool for possible conflicts. RAK13600 uses I2C and IO pins. It can cause possible conflict, especially on some IO modules.

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

- Battery 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.
- Make sure the battery wires match the polarity on the RAK WisBlock Base Board. Not all batteries have the same wiring.
- 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.

Software Configuration and Example

The RAK13600 is based on the popular NFC/RFID chip PN532. You need to install the RAK13600-PN532 library to use the module. By default, the module communicates via I2C to the WisBlock Core module. The following guide shows how to test your RAK13600 module using a standard RFID card, as shown in **Figure 7**.



Figure 7: RAK13600 with Coil and RFID Card

1. Open the Arduino IDE and select the WisBlock Core you use, as shown in Figure 8 to Figure 10.

💿 sketch_aug22b .	Arduino 1.8.15						_	đ	\times
File Edit Sketch Too	ols Help								
	Auto Format	Ctrl+T							<u>.</u>
	Archive Sketch								
sketch_aug22	Fix Encoding & Reload								
1 void s	Manage Libraries	Ctrl+Shift+I							^
2 // p	Serial Monitor	Ctrl+Shift+M	nce:						
3	Serial Plotter	Ctrl+Shift+L							
4 }	WiFi101 / WiFiNINA Firmware Updater								
5	Board: "WicBlock Core RAK4631 Board"	2	Boards Manager						
6 void 1	Bootloader: "0.3.2 SoftDevice s140.6.1.1	" >	Arduino AVR Boards	>					
7 // p	Debug: "Level 0 (Release)"		Arduino SAMD (32-bits ARM Cortex-M0+) Boards	Ś					
8	Port: "COM4"	>	ESP8266 Boards (3.0.2)	>					
91	Get Board Info		ESP8266 Boards (3.1.0-dev)	>					
5,1			RAKwireless ESP32 Modules	>					
	Programmer	>	RAKwireless nRF Modules	•	WisBlock Core RAK4631 Board				
	Burn Bootloader				WisBlock Core RAK4601 Board				
									~
7					WisBlock Co	ore RAK4631 Board. 0.3.2 SoftDevice s140 6	.1.1. Level 0 (Re	lease) on C	OM4

Figure 8: Selecting RAK4631 as WisBlock Core

sketch_aug22c Arduino 1.8.15					-	đ	×
File Edit Sketch Tools Help							
Auto Format		Ctrl+T					9 -
Archive Sketch							_
sketch_aug2: Fix Encoding & Reload							
1 void s Manage Libraries		Ctrl+Shift+I					^
2 // D Serial Monitor		Ctrl+Shift+M					
Serial Plotter		Ctrl+Shift+L					
4 } WiFi101 / WiFiNINA Firmw	vare Updater						
5 Board: "WisCore RAK1120	0 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 RAK11200 Board, De	efault 4MB with spiffs (1.2MB APP/1.5MB	SPIFFS), QIO, 80MHz, 921600), None on CO	DM4

Figure 9: Selecting RAK11200 as WisBlock Core

∞ sketch_oct29a /	Arduino 1.8.15					_	Ø	\times
File Edit Sketch To	ols Help							
	Auto Format	Ctrl+T						.0
	Archive Sketch							
sketch oct29	Fix Encoding & Reload							-
1 word a	Manage Libraries	Ctrl+Shift+I						^
	Serial Monitor	Ctrl+Shift+M						
2 // p	Serial Plotter	Ctrl+Shift+L	once:					
3								
4 }	WiFi101 / WiFiNINA Firmware Update	r						
5	Board: "WisBlock RAK11300"	>	Boards Manager					
6 void l	Port: "COM10 (WisBlock RAK11300)"	>	Arduino AVR Boards	>				
7 // p	Get Board Info		Arduino SAMD (32-bits ARM Cortex-M0+) Boards	>				
8	D	,	ESP8266 Boards (3.0.2)	>				
9 }	Programmer		ESP8266 Boards (3.1.0-dev)	>				
	Burn Bootloader		RAKwireless ESP32 Modules	>				
			RAKwireless nRF Modules	>				
			Rakwireless Raspberry Modules	> • WisBlock RAK11300				
								~
1					- V	VISBIOCK RAK1	1300 on CC	UTMC

Figure 10: Selecting RAK11300 as WisBlock Core

2. Copy the example code below and paste it on the Arduino IDE:

BAK[°] Documentation Center

```
@file readMifareClassic.ino
  @author rakwireless.com
  @brief This example will wait for any ISO14443A card or tag, and depending on the size of the
#include <Wire.h>
#include <SPI.h>
#include <RAK13600-PN532.h> // Click here to get the library: http://librarymanager/All#RAK13600
#define PN532_IRQ (WB_I06)
#define PN532_RESET (WB_I05) // Not connected by default on the NFC Shield
NFC_PN532 nfc(PN532_IRQ, PN532_RESET);
void setup(void) {
 Serial.begin(115200);
 pinMode(WB_I02, OUTPUT);
 digitalWrite(WB_I02, HIGH);
 delay(300);
 while (!Serial) delay(10); // for Leonardo/Micro/Zero
 Serial.println("Hello!");
 nfc.begin();
 uint32_t versiondata = nfc.getFirmwareVersion();
 if (! versiondata) {
   Serial.print("Didn't find PN53x board");
   while (1); // halt
 Serial.print("Found chip PN5"); Serial.println((versiondata>>24) & 0xFF, HEX);
 Serial.print("Firmware ver. "); Serial.print((versiondata>>16) & 0xFF, DEC);
 Serial.print('.'); Serial.println((versiondata>>8) & 0xFF, DEC);
 nfc.SAMConfig();
 Serial.println("Waiting for an IS014443A Card ...");
void loop(void) {
 uint8_t success;
 uint8_t uid[] = { 0, 0, 0, 0, 0, 0, 0 }; // Buffer to store the returned UID
 uint8_t uidLength;
                                            // Length of the UID (4 or 7 bytes depending on ISO14
     if the uid is 4 bytes (Mifare Classic) or 7 bytes (Mifare Ultralight)
 success = nfc.readPassiveTargetID(PN532_MIFARE_IS014443A, uid, &uidLength);
 if (success) {
   Serial.println("Found an IS014443A card");
   Serial.print(" UID Length: ");Serial.print(uidLength, DEC);Serial.println(" bytes");
   Serial.print(" UID Value: ");
```

```
nfc.PrintHex(uid, uidLength);

if (uidLength == 4)
{
 // We probably have a Mifare Classic card ...
 uint32_t cardid = uid[0];
 cardid <<= 8;
 cardid |= uid[1];
 cardid <<= 8;
 cardid |= uid[2];
 cardid <<= 8;
 cardid |= uid[3];
 Serial.print("Seems to be a Mifare Classic card #");
 Serial.println(cardid);
 }
 Serial.println("");
}
delay(2000);
</pre>
```

(H)

3. Install the RAK13600-PN532 library by clicking the link highlighted by red box, as shown in **Figure 11**.

💿 sketch_feb15a Arduino 1.8.16 —		×	
		Ø	
sketch_feb15a §			
<pre>/** @file readMifareClassic.ino @author rakwireless.com @brief This example will wait for any ISO14443A card or tag, and depending on the size of the UID will attempt to read from it. @version 0.1 @date 2021-10-14 @copyright Copyright (c) 2021 **/</pre>			^
<pre>#include <wire.h> #include <spi.h> #include <spi.h> #include <rak13600-pn532.h> // Click here to get the library: http://librarymanager/All#RAK13600-PN532</rak13600-pn532.h></spi.h></spi.h></wire.h></pre>			
<pre>// If using the preakout or shield with 12C, define just the pins connected #define PN532_IRQ (WB_IO6) #define PN532_RESET (WB_IO5) // Not connected by default on the NFC Shield</pre>			
<pre>// Or use this line for a breakout or shield with an I2C connection: NFC_PN532 nfc(PN532_IRQ, PN532_RESET);</pre>			
<pre>void setup(void) { Serial.begin(115200); pinMode(WB_IO2, OUTPUT); digitalWrite(WB_IO2, HIGH); delay(300);</pre>			
<pre>while (!Serial) delay(10); // for Leonardo/Micro/Zero</pre>			v
27 WisBlock RAK4831, S140 6.1.1, Level 0 (Rele	ase) on C	0М25	

Figure 11: Getting the RAK13600-PN532 Library

4. You will be directed to the Library Manager then you have to click install.

🐵 sketch_feb15a | Arduino 1.8.16

_	\sim

File Edit Sketch Tools Help		
		Ø
sketch_feb15a§		
/** @ Library Manager @author rakwi @ Library Manager @brief This Type All Topic All KAK13600-PN532 @version 0.1 @ RAKwireless RAK13600 RFID library @date 2021-10 by RAKWireless @copyright Col BrackWireless library for the RAK13600 RFID module with the PN532 RFID reader and writer module	from RAKWireless.	t .
<pre>Checkout the <u>WisBlock RAK13600</u> today! More info finclude <spi.h finclude <rak136 // If using the fdefine PN532_IR fdefine PN532_RR</rak136 </spi.h </pre>	Install	
<pre>NFC_PN532 nfc(PN void setup(void) Serial.begin(1 pinMode(WB_IO2 digitalWrite(W delay(300); while (!Serial</pre>	Close	v
12	WisBlock RAK4631, S140 6.1.1, Level 0 (F	Release) on COM25

Figure 12: Installing RAK13600-PN532 Library

5. Next, you need to install an additional library from Adafruit. You need to type on the library search box adafruit busio then install it by clicking install. After the successful installation of the two libraries, you can now close the Library Manager window.

File Edit Sketch Tools Help	😳 sketch_feb15a Arduino 1.8.16		— 🗆 X
<pre>i i i i i i i i i i i i i i i i i i i</pre>	File Edit Sketch Tools Help		
sketch jeb133 // If using the fdefine PIS32_IF fdefine PIS32_IF fdefine PIS32_R // Or use this i NFC_PIS32 ntcr{N} Serial.printlr ntc.begin(); uhit2(1);;) // Oct ok data (Version 1:1:1 v Install (Version 1:1:1 v Install) (Version 1:1:1 v Install version vers			P
<pre>// If using the fdefine PIS32_R fdefine PIS32_R // Or use this NFC_PIS32 ntc(F) void setup(void) by Adafuit This is a library for abstracting away UART, 12C and SPI interfacing This is a library for abstracting away UART, 12C and SPI interfacing Normality void setup(void) Serial.peinter ffc.begin(); uint32_t versa if (! versaind Serial.print while (1); } } // Got ok data</pre>	sketch_feb15a		
<pre>// If Using the Correct Wanager // If Using the Correct Wanager // If Using the PN532 If define PN532 If the I brand for the PN532 If define PN532 If the I brand for the</pre>			^
<pre>tdefine H0532_R fupe M vpe M vpe A vp</pre>	// II using the w Library Manager	^	
Adafruit BusIO NFC_BNS32 nfc (R) vid setup (void) Serial.begin () pimbode (MB_DOG digitalWrite (indexing) Were info Version 1.11.1 \rgc Install infc.begin (); uint32_t versi if (! versiond Serial.printing while (1); j // Got ok date	#define PN532_RE Type All V Topic All V adafruit busio		
<pre>// Or use this] NFC_PNS32 nfc(Pi NFC_PNS32 nfc(Pi Fis is alibrary for abstracting away UART, 12C and SPI interfacing This is a library for abstracting away UART, 12C and SPI This is a library for abstracting away UART, 12C and SPI Nore info Serial.egin() if (! version Serial.print while (1); } // Got ok date Core 1 UBELOR KAK491, S14D 0.11, Level 0.Release) on COM25 </pre>	Adafruit BusIO		
Integradue This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting away UART. I2C and SPI interfacing This is a library for abstracting the set interfacing This is a library for abstracting the set interfacing This is a library for abstracting the set interfacing This is a library for abstracting the set interfacing This is a library for abstracting the set interfacing This is a library for abstracting the set interfacing the set inte	// Or use this 1 NEC DNE22 pfo(DN by Adafruit		
void setup (void) Mare info Serial.begin (1) Install digitalWrite (# delay(300); while (!Serial Serial.println nfc.begin (); uint32_f.versi if (!version serial.println while (1); /	This is a library for abstracting away UART, I2C and SPI interfacing This is a library for abstracting away UART, I2C and SPI interfacing		
<pre>serial.begin() if (! version Serial.printl nfc.begin(); iint32_t versi if (! version Serial.print while (1); } // Got ok data Close </pre>	void setup (void) More info	_	
<pre>pinMode(WB_IO2 digitalWrie(W delay(300); while (!Serial Serial.printlr nfc.begin(); uint32_t versi if (! version Serial.print while (1); } // Got ok date Close 1 WhElock RAK4831, S14D 8.1.1, Level D (Release) on COM25</pre>	Serial.begin (1		
digitalWrite(k delay(300); while (iSerial Serial.printlr nfc.begin(); uint32_t versi if (! versiond Serial.print while (1); / // Got ok data It WisBlock RAM-4831, S140 0.1.1, Level 0 (Release) on COM25	pinMode (WB_IO2		
<pre>delay(300); while (!Serial Serial.println nfc.begin(); uint32_t versi if (! versiond Serial.print while (1); / } // Got ok data Close </pre>	digitalWrite (M	-	
<pre>serial.println nfc.begin(); uint32_t versi if (! versiond Serial.print while (1); / } // Got ok data Close 1 WisBlock RAK-4031, S140 .6.1.1, Level 0 (Release) on COM25</pre>	delay(suu);		
Serial.printlr nfc.begin(); uint32_t versi if (! versiond Serial.print while (1); / } // Got ok data Close 1 WisBlock RAK4031, S140 0.1.1, Level 0 (Release) on COM25	white (
<pre>nfc.begin(); uint32_t versi if (! versiond Serial.print while (l); / } // Got ok data Close</pre>	Serial.println		
uint32_t versi if (! versiond Serial.print while (1); / } // Got ok data Close 1 WisBlock RAK4031, S140 6.1.1, Level D (Release) on COM25	nfc.begin();		
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25	uint32_t versi		
Serial.print while (1); / } // Got ok data Close 1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25	if (! versiond		
while (1); / // Got ok data // Got ok data Close 1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25	Serial.print		
) // Got ok data Close 1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25	while (1); /		
I WisBlock RAK4631, S140 6.1.1, Level 0 (Release) on COM25	// Cot ok data	~	
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25	// GOC OX data	Close	
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25			
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25			
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25			
1 WisBlock RAK4631, S140 6.1.1, Level O (Release) on COM25			
1 WisBlock RAK4831, S140–6.1.1, Level D (Release) on COM25			
1 WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25			
	1 WisBlock RAK4831, S140 6.1	.1, Level 0	(Release) on COM25

Figure 13: Installing Adafruit BusIO Library

6. Select the right Serial Port and upload the code, as shown in Figure 14 and Figure 15.

sketch_feb15a

Ofile read

@author ra

Obrief Th

Qversion (

×

Ø

@date 2021 @copyright Board: "WisBlock RAK4631" SoftDevice: "S140 6.1.1" #include <Wir</pre> Debug: "Level 0 (Release)" #include <SPI</pre> Port: "COM25" #include <RAK manager/All#RAK13600-PN532 Get Board Info COM25 // If using t Programmer #define PN532

Burn Bootloader #define PN532 the NFC Shield

// Or use this line for a breakout or shield with an I2C connection: NFC_PN532 nfc(PN532_IRQ, PN532_RESET);

<pre>while (!Serial) delay(10); // for Leonardo/Micro/Zero</pre>	<pre>void setup(void) { Serial.begin(115200); pinMode(WB_IO2, OUTPUT) digitalWrite(WB_IO2, HI delay(300); while (!Serial) delay(1</pre>	: SH);)); // for Leonardo/Micro/Zero	

Figure 14: Selecting the correct Serial Port

🥯 sketch_feb15a | Arduino 1.8.16 \times _ File Edit Sketch Tools Help Ø sketch_feb15a if (success) { // Display some basic information about the card Serial.println("Found an ISO14443A card"); Serial.print(" UID Length: ");Serial.print(uidLength, DEC);Serial.println(" bytes"); Serial.print(" UID Value: "); nfc.PrintHex(uid, uidLength); if (uidLength == 4) { // We probably have a Mifare Classic card ... uint32_t cardid = uid[0]; cardid <<= 8; cardid |= uid[1]; cardid <<= 8; cardid |= uid[2]; cardid <<= 8; cardid |= uid[3]; Serial.print("Seems to be a Mifare Classic card #"); Serial.println(cardid); Serial.println(""); delav(2000); loading WisBlock RAK4831, S140 6.1.1, Level 0 (Release) on COM25

Figure 15: Uploading the sample code

If you experience any error in compiling the demo sketch, check the updated code for the RAK13600 readMifareClassic example 🖸 . Other example codes for RAK13600 are also available on the RAK13600 WisBlock Example Code Repository \square .

NOTE:

If you are using RAK11200 as the WisBlock Core, it 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.

7. When you have successfully uploaded the example code, open the serial monitor by clicking the magnifying glass icon on the upper right of the Arduino IDE then place the RFID card on top of the antenna coil. You should see the details of the card, as shown in **Figure 16**.

🥯 sket	tch_feb15a Arduino 1.8.16			
File Edit	it Sketch Tools Help		_	_
ØC			P	
sketc	© COM25 —			
if (Send		
// Se Se If	Hello! Found chip PN532 Firmware ver. 1.6 Waiting for an ISO14443A Card Found an ISO14443A card UID Length: 4 bytes UID Value: 0x3F 0xAD 0x8D 0x29 Seems to be a Mifare Classic card #1068338473			
} Se } dela }				v
Activat Device	Autoscroll Show timestamp Both NL & CR 115200 baud programmed.	Clear output		^
DFU upg	grade took 3.7088732719421387s		>	~
1	WisBlock RAK4831, S140, 6,1	1. Level 0 (Release) on COM25	

Figure 16: RFID Card Information Read by the RAK13600

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

RAK13600 WisBlock NFC Reader Module Datasheet

Overview

Description

RAK13600 is a **WisBlock Wireless** NFC reader module based on the PN532 chip. It provides a compact and lowpower solution for NFC-related IoT applications. The RAK13600 has an integrated transceiver module used for NFC applications which operates at 13.56 MHz and is also based on the standard 80C51 microcontroller core. It supports 6 different operating modes:

- ISO/IEC 14443A/MIFARE Reader/Writer
- FeliCa Reader/Writer
- ISO/IEC 14443B Reader/Writer
- ISO/IEC 14443A/MIFARE Card MIFARE Classic 1K or MIFARE Classic 4K card emulation mode
- FeliCa Card emulation
- ISO/IEC 18092, ECMA 340 Peer-to-Peer

Features

- Module specifications
 - Based on NXP PN532
 - External Antenna
 - Low power
- Size
 - 35 x 25 mm

Specifications

Overview

Mounting

The RAK13600 NFC reader module can be mounted to the IO slot of the WisBlock Base board. To illustrate, **Figure 1** shows how RAK13600 can be mounted on RAK5005-O WisBlock Base.



Figure 1: RAK13600 Mounting to WisBlock Base

Hardware

The hardware specification is categorized into four parts. It discusses the pinouts of the module and its corresponding functions. It also covers the electrical and mechanical parameters that include the tabular data of its characteristics. This section also shows the schematic diagram of RAK13600.

Chipset

Vendor	Part number
NXP	PN532

Pin Definition

The RAK13600 WisBlock NFC Reader has the standard 40-pin WisConnector. It is compatible with the IO slot which allows the RAK13600 module to be mounted on a WisBlock baseboard, such as RAK5005-O. The pin order of the connector and the Pinout Definition is shown in **Figure 2**.



Figure 2: RAK13600 IO Slot Connector

NOTE:

- The default interface is **I2C**.
- Reset functionality is available on **RESET (IO5)** and **RSTPD_N (IO4)** pins.
- PN532 wake up is available on WAKEUP (IO6).
- **3V3_S** and **GND** supply power to the board.

Electrical Characteristics

BAK[°] Documentation Center

Parameter	Minimum	Typical	Maximum	Unit
3V3_S	2.7	3.3	3.6	V
I _{dvdd} (Digital Supply Current)		25		mA
I _{avdd} (Analog Supply Current)		6		mA
I _{tvdd} (Transmitter Supply Current)		60	150	mA
Ambient Temperature	-30		80	°C

Mechanical Characteristics

Board Dimensions

Refer to **Figure 3** below for the mechanical dimensions of the RAK13600 module.



Figure 3: RAK13600 Mechanical Dimensions

WisConnector PCB Layout



Figure 4: WisConnector PCB footprint and recommendations







Certification

Last Updated: 8/5/2022, 4:34:30 AM