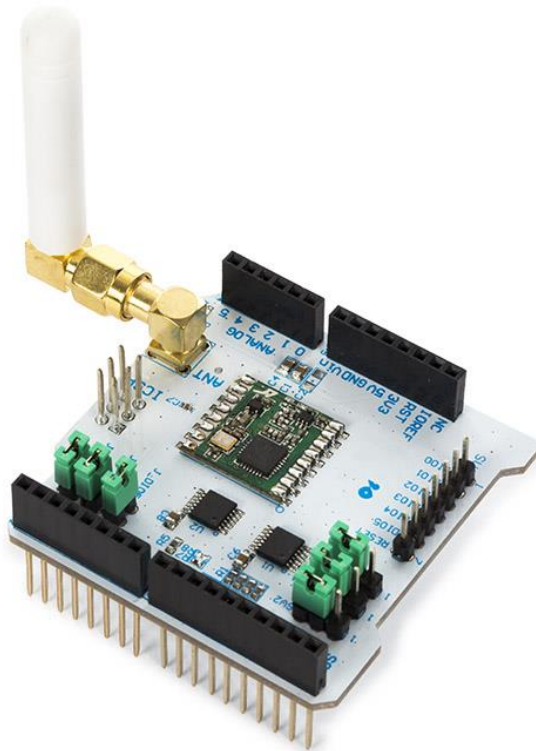


VMA214



USER MANUAL

1. Introduction

To all residents of the European Union



Important environmental information about this product

This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.

If in doubt, contact your local waste disposal authorities.

Thank you for choosing Velleman! Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

2. Safety Instructions



Read and understand this manual and all safety signs before using this appliance.



Indoor use only.

- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.

3. General Guidelines

- Refer to the Velleman® Service and Quality Warranty on the last pages of this manual.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorised way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- Nor Velleman nv nor its dealers can be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical...) arising from the possession, use or failure of this product.
- Keep this manual for future reference.

4. What is Arduino®

Arduino® is an open-source prototyping platform based in easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing).

Surf to www.arduino.cc and www.arduino.org for more information.

5. Overview

General

The RFM69HCW module is an inexpensive and versatile radio module. You can use it to send text or binary data between two or hundreds of modules. It is perfect for building inexpensive short-range wireless networks for home and building automation, sensor networks, automated meter readings, wireless alarm and security systems, industrial monitoring and control, and many more applications!

The RFM69HCW transceiver module operates over a wide frequency range, including the 315, 433, 868 and 915 MHz license-free ISM (Industry Scientific and Medical) frequency bands. All major RF communication parameters are programmable and most can be set dynamically. The RFM69HCW is optimized for low power consumption while offering high RF output power and channelized operation.

This VMA214 RFM69HCW Radio Shield connects the RFM69HCW module to the appropriate lines on the Arduino®. By using the RF69 Library, you can send and receive messages via standard 4-wire SPI interface. The library includes command structures for setting up anything from simple non-addressed point-to-point communication to fully addressed networks of clients and routers. Frequency can be set with 61 Hz precision to any frequency from 240.0 MHz to 960.0 MHz.

The shield has on-board power regulation and level shifting, an external antenna via the I-Pex connector and additional jumpers to set the interface and digital communication pins on the Arduino®.

Technical Specifications

168 dB maximum link budget

+20 dBm - 100 mW constant RF output

high sensitivity down to -120 dBm at 1.2 kbps

high selectivity 16-tap FIR channel filter

low current Rx = 16 mA, 100 mA register retention

programmable Pout -18 to +20 dBm in 1 dB step

FSK bit rates up to 300 kb/s

fully integrated synthesizer with a resolution of 61 Hz

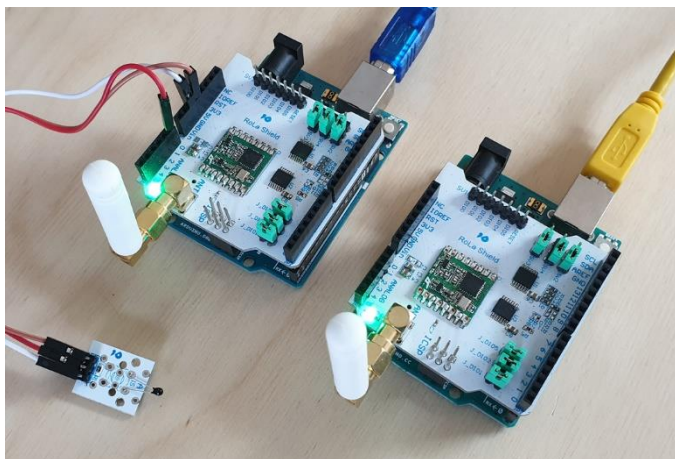
FSK, GFSK, MSK, GMSK, LoRa™ and OOK modulation

packet engine with CRC-16, AES-128, 66-byte FIFO

115 dB+ dynamic range RSSI

6. Operation

Overview



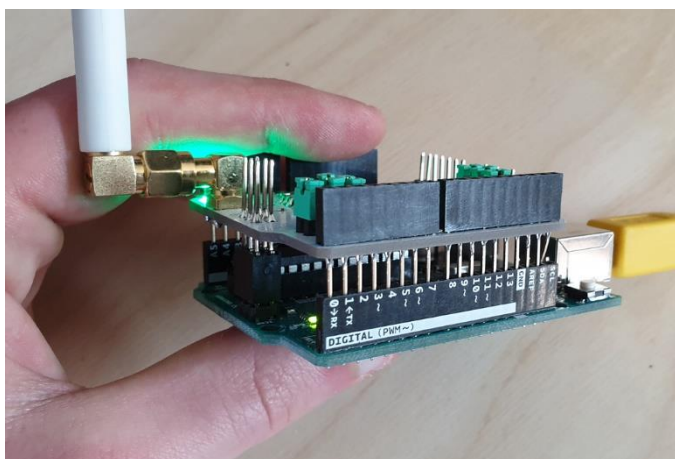
We will connect two radio shields to one another, read the temperature and send the temperature value to another module.

Requirements:

- 2x Arduino® UNO
- 2x VMA214 RFM69HCW module
- 1x VMA320 analogue temperature sensor module
- 2x USB-A to USB-B cable

Connecting Arduino®

1. Connect the VMA214 RFM69HCW module to the Arduino® UNO (or Arduino® MEGA) as depicted.

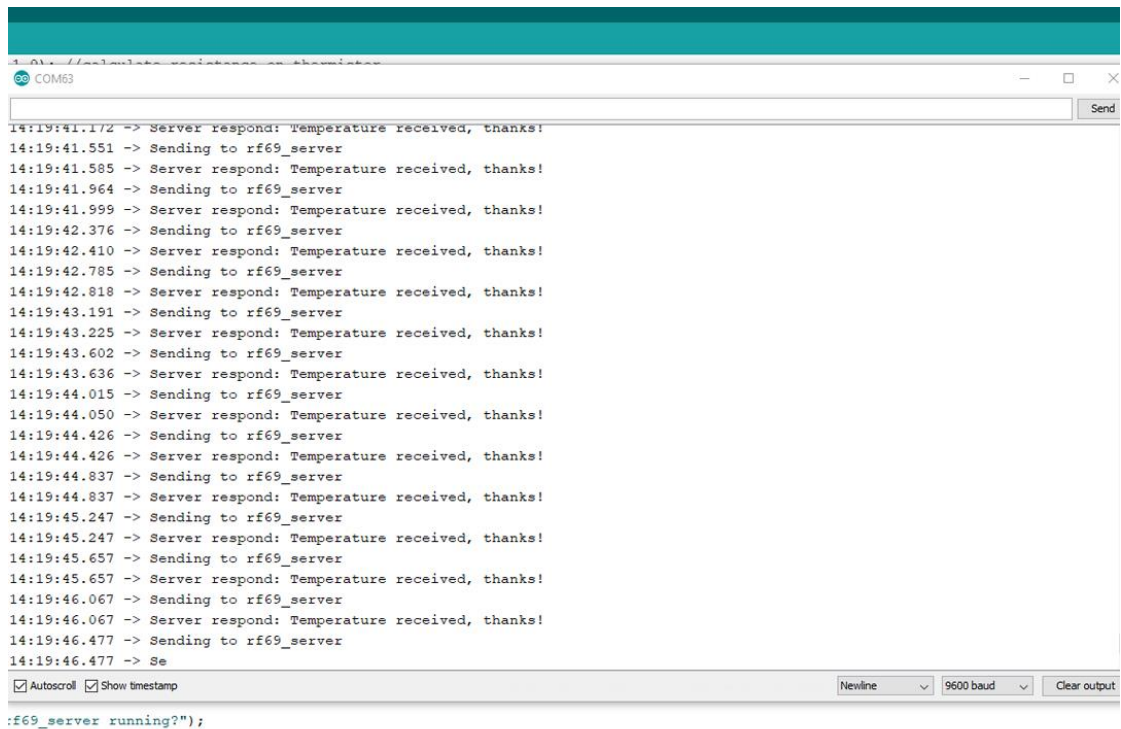


2. Connect the VMA320 temperature module.

VMA320		VMA214
+	→	5V
-	→	GND
S	→	A0

Programming

1. Download the RadioHead library [here](#) and add it to the Arduino® IDE.
2. Go to GitHub and download both codes [here](#).
3. Open the RF69_CLIENT_SEND_DATA code and upload to the Arduino® **with** the temperature sensor. You should see following output:



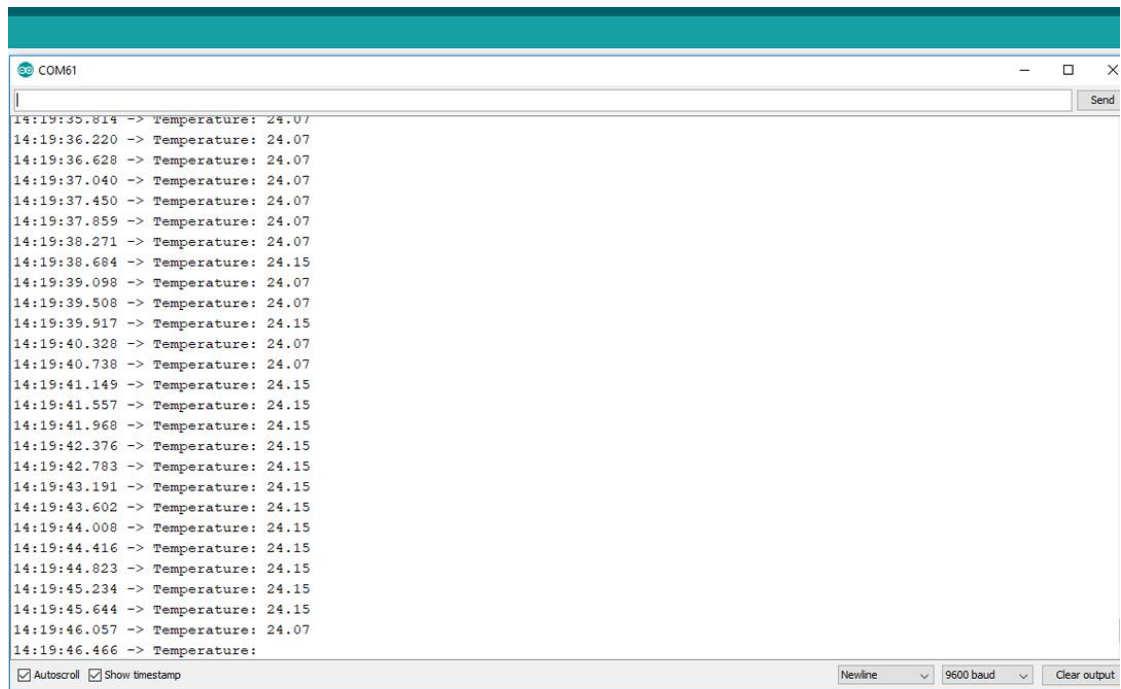
```

COM63
14:19:41.172 -> Server respond: Temperature received, thanks!
14:19:41.551 -> Sending to rf69_server
14:19:41.585 -> Server respond: Temperature received, thanks!
14:19:41.964 -> Sending to rf69_server
14:19:41.999 -> Server respond: Temperature received, thanks!
14:19:42.376 -> Sending to rf69_server
14:19:42.410 -> Server respond: Temperature received, thanks!
14:19:42.785 -> Sending to rf69_server
14:19:42.818 -> Server respond: Temperature received, thanks!
14:19:43.191 -> Sending to rf69_server
14:19:43.225 -> Server respond: Temperature received, thanks!
14:19:43.602 -> Sending to rf69_server
14:19:43.636 -> Server respond: Temperature received, thanks!
14:19:44.015 -> Sending to rf69_server
14:19:44.050 -> Server respond: Temperature received, thanks!
14:19:44.426 -> Sending to rf69_server
14:19:44.426 -> Server respond: Temperature received, thanks!
14:19:44.837 -> Sending to rf69_server
14:19:44.837 -> Server respond: Temperature received, thanks!
14:19:45.247 -> Sending to rf69_server
14:19:45.247 -> Server respond: Temperature received, thanks!
14:19:45.657 -> Sending to rf69_server
14:19:45.657 -> Server respond: Temperature received, thanks!
14:19:46.067 -> Sending to rf69_server
14:19:46.067 -> Server respond: Temperature received, thanks!
14:19:46.477 -> Sending to rf69_server
14:19:46.477 -> Se

Autoscroll Show timestamp Newline 9600 baud Clear output
:rf69_server running?");

```

4. Open a new IDE with the RF69_SERVER_RECEIVE_DATA code and upload to the Arduino® **without** the temperature sensor. You should see following output:



```

COM61
14:19:35.814 -> Temperature: 24.07
14:19:36.220 -> Temperature: 24.07
14:19:36.628 -> Temperature: 24.07
14:19:37.040 -> Temperature: 24.07
14:19:37.450 -> Temperature: 24.07
14:19:37.859 -> Temperature: 24.07
14:19:38.271 -> Temperature: 24.07
14:19:38.684 -> Temperature: 24.15
14:19:39.098 -> Temperature: 24.07
14:19:39.508 -> Temperature: 24.07
14:19:39.917 -> Temperature: 24.15
14:19:40.328 -> Temperature: 24.07
14:19:40.738 -> Temperature: 24.07
14:19:41.149 -> Temperature: 24.15
14:19:41.557 -> Temperature: 24.15
14:19:41.968 -> Temperature: 24.15
14:19:42.376 -> Temperature: 24.15
14:19:42.783 -> Temperature: 24.15
14:19:43.191 -> Temperature: 24.15
14:19:43.602 -> Temperature: 24.15
14:19:44.008 -> Temperature: 24.15
14:19:44.416 -> Temperature: 24.15
14:19:44.823 -> Temperature: 24.15
14:19:45.234 -> Temperature: 24.15
14:19:45.644 -> Temperature: 24.15
14:19:46.057 -> Temperature: 24.07
14:19:46.466 -> Temperature:

Autoscroll Show timestamp Newline 9600 baud Clear output

```

Now, it's time to try and send data using all kinds of sensors, e.g. reading and sending the status of a PIR sensor to the Arduino® so you can make a relay switch or control LEDs...

Use this device with original accessories only. Velleman nv cannot be held responsible in the event of damage or injury resulting from (incorrect) use of this device. For more info concerning this product and the latest version of this manual, please visit our website www.velleman.eu. The information in this manual is subject to change without prior notice.

© COPYRIGHT NOTICE

The copyright to this manual is owned by Velleman nv. All worldwide rights reserved.
No part of this manual may be copied, reproduced, translated or reduced to any electronic medium or otherwise without the prior written consent of the copyright holder.

Velleman® Service and Quality Warranty

Since its foundation in 1972, Velleman® acquired extensive experience in the electronics world and currently distributes its products in over 85 countries.

All our products fulfil strict quality requirements and legal stipulations in the EU. In order to ensure the quality, our products regularly go through an extra quality check, both by an internal quality department and by specialized external organisations. If, all precautionary measures notwithstanding, problems should occur, please make appeal to our warranty (see guarantee conditions).

General Warranty Conditions Concerning Consumer Products (for EU):

- All consumer products are subject to a 24-month warranty on production flaws and defective material as from the original date of purchase.
- Velleman® can decide to replace an article with an equivalent article, or to refund the retail value totally or partially when the complaint is valid and a free repair or replacement of the article is impossible, or if the expenses are out of proportion.

You will be delivered a replacing article or a refund at the value of 100% of the purchase price in case of a flaw occurred in the first year after the date of purchase and delivery, or a replacing article at 50% of the purchase price or a refund at the value of 50% of the retail value in case of a flaw occurred in the second year after the date of purchase and delivery.

• Not covered by warranty:

- all direct or indirect damage caused after delivery to the article (e.g. by oxidation, shocks, falls, dust, dirt, humidity...), and by the article, as well as its contents (e.g. data loss), compensation for loss of profits;
- consumable goods, parts or accessories that are subject to an aging process during normal use, such as batteries (rechargeable, non-rechargeable, built-in or replaceable), lamps, rubber parts, drive belts... (unlimited list);
- flaws resulting from fire, water damage, lightning, accident, natural disaster, etc....;
- flaws caused deliberately, negligently or resulting from improper handling, negligent maintenance, abusive use or use contrary to the manufacturer's instructions;
- damage caused by a commercial, professional or collective use of the article (the warranty validity will be reduced to six (6) months when the article is used professionally);
- damage resulting from an inappropriate packing and shipping of the article;
- all damage caused by modification, repair or alteration performed by a third party without written permission by Velleman®.
- Articles to be repaired must be delivered to your Velleman® dealer, solidly packed (preferably in the original packaging), and be completed with the original receipt of purchase and a clear flaw description.
- Hint: In order to save on cost and time, please reread the manual and check if the flaw is caused by obvious causes prior to presenting the article for repair. Note that returning a non-defective article can also involve handling costs.
- Repairs occurring after warranty expiration are subject to shipping costs.
- The above conditions are without prejudice to all commercial warranties.

The above enumeration is subject to modification according to the article (see article's manual).