Live USB instructions:

This is a guide for how to install the necessary tools to run a hello-world program on avr-rss2 microcomputer. It requires that you either have an Ubuntu machine, a VM, or a live USB with the Ubuntu OS. Ask Erik for a working VM folder (5.7 GB) to use in VMware Player.

Plug in the USB into your computer and get Ubuntu to run (a special key has to be pressed when starting the computer).

Get internet connection. Change keyboard layout to Swedish see:
https://www.wikihow.com/Change-Keyboard-Layout-in-Ubuntu

Go into the terminal and run

sudo apt-get update

followed by

sudo apt-get install gcc-avr avr-libc avrdude make

You also have to clone to git repository

sudo git clone --recursive https://github.com/team-ethernet/contiki.git

Then relocate yourself to the Hello-world directory

cd contiki/examples/hello-world

Compiling the code

Now you are ready to compile the program. Make sure that you have plugged in the microcomputer into an USB-port on your computer before this.

sudo make TARGET=avr-rss2

Running the code

Now you are ready to run the program with avrdude. However you might have to find the USB-port used for the microcomputer.
Our solution was to use the console command

ls /dev

before and after disconnecting and reconnecting the USB to see which port appeared and disappeared in the list of all ports currently being used.

When you have found which one is being used, edit the **blue text** to match your port in the command below. Before running the code press down the reset button on the

microcomputer (it is gray) and release. While the LED is flashing run the code in the terminal. If you are too slow or to fast you might miss the opportunity and get a timeout message.

sudo avrdude -p m256rfr2 -c stk500v2 -P /dev/ttyUSB0 -b 115200 -e -U flash:w:hello-world.avr-rss2

If you can not see the USB then restarting the VM with the USB connected to the computer could solve the problem.

To see the output, use Putty in Windows or an other serial program such as minicom. Below follows instructions for setting up and using minicom.

Minicom setup

To use minicom, first install it with the command in the terminal

sudo apt-get install minicom

Then run setup with the command

sudo minicom -s

In setup, choose “Serial port setup” navigating yourself using the **arrow keys** and press **Enter**.
Once done press the **A-key** to select the “Serial device option” and change the serial device to

/dev/ttyUSB0

Then press **Enter** once follow by the **E-key** to select the “BPS/Par/Bits”-section. When in that section press the **D-key** followed by **Enter** to change the speed to 38400 and then **Enter** again to leave the “Serial port setup” section.

Once this is done you use the **arrow keys** again to select the “Save setup as dfl” where dfl stands for default. Press **Enter** to select it, wait a few seconds and congratulations you are done \(^^)/.

Now use the **arrow-keys** to navigate yourself to the “Exit”-section and press **Enter** to start the communication.

Exiting running minicom session

To exit a running minicom session press **Ctrl+a** release this combination quickly followed by the button **Z** then **X** then **Enter** to exit the running minicom session.

Starting Minicom again

Since you saved the settings as default the next time you would like to start minicom you only have to type

sudo minicom

To compile and run your own program

To compile and run your own program change the directory to the directory where your program is. Compile it by using the

sudo make TARGET=avr-rss2

Assuming you already know the port you can then repeat the steps in **Running the code**. You will however have to change the name marked with **red text** below with the name of your program.

sudo avrdude -p m256rfr2 -c stk500v2 -P /dev/ttyUSB0 -b 115200 -e -U flash:w:hello-world.avr-rss2

To run the MQTT-Broker on your computer

Firstly you will have to download mosquitto
On windows:
See separate guide

On Mac:
We download mosquitto using brew with this command and you can position yourself in the main directory

brew install mosquitto

If you have some problems these two commands could be needed

brew link mosquitto

and

brew upgrade mosquitto

Then to start your broker you first have to write the magical command

ln -sfv /usr/local/opt/mosquitto/*.plist ~/Library/LaunchAgents

and then to launch type

launchctl load ~/Library/LaunchAgents/homebrew.mxcl.mosquitto.plist

or

brew services start mosquitto

If you want to restart the broker or stop it you can use these commands

brew services restart mosquitto
brew services stop mosquitto

Congratulations you now have the broker up and running!

To see if it works you can open two new terminal windows and subscribe to it and publish to the broker.

To subscribe type in one terminal window

mosquitto_sub -h 127.0.0.1 -t noisesensor/+/sensors

To publish type in one terminal window

mosquitto_pub -h 127.0.0.1 -t topic -m "Hello World"

If everything works you should now see a “Hello World” text pop up in the subscriber terminal window.

Communication between noise sensor and your broker

This guides assumes you know how to compile and run a program from a Linux system or VM to your avr-rss2. If you do not know how to do this follow the “How to compile and run your own program” guide above.

Using text editor

To make the microcomputer communicate with your bridge you first have to change the IP:address of the broker it sends data to. To do this you can use a simple text editor and in the directory **contiki/examples/avr-rss2/ipv6/mqtt-noisesensor** edit the file project-conf.h at row 84 (or close to this one). What you need to change are the last 4 bytes of the MQTT_DEMO_BROKER_IP_ADDR row. The last 4 bytes should be your IP-address encoded in hexadecimal code like this

MQTT_DEMO_BROKER_IP_ADDR "0064:ff9b::82e5:881f" //130.229.136.31

Using Nano

Navigate yourself to the mqtt-noisesensor project by

cd contiki/examples/avr-rss2/ipv6/mqtt-noisesensor

and type in the terminal

sudo nano project-conf.h

and change the last four bytes of the MQTT_DEMO_BROKER_IP_ADDR (you will have to scroll down quite a bit) in the project-conf.h file to the hexadecimal encoded IP address to the broker that will be used. Example below.

MQTT_DEMO_BROKER_IP_ADDR "0064:ff9b::82e5:881f" //130.229.136.31

Exit by pressing CTRL+X, selecting Yes (Y), and then pressing ENTER to confirm file name.