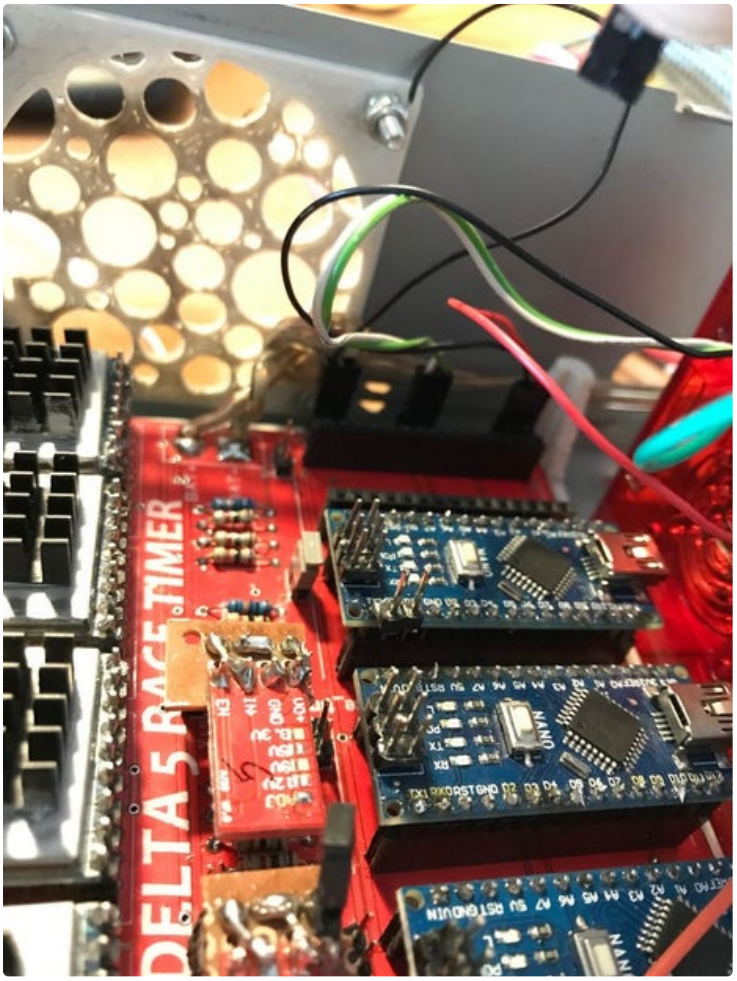
RotorHazard OTA Updater

by szafranski

Collect all of required components.
Make sure that you have internet connection.

Supplies:

1. RotorHazard race timer
2. About 20 cm of thin wire + jumper wires or tool for attaching female gold-pins to the wire
3. Soldering iron + solder
4. 2 resistors: 5kOhm and 10 kOhm - or any other combination with 1/2 ratio in 5-20 kOhm range OR logic level converter
5. PC connected to the internet



Step 1: Make Additional PCB Wiring

Connect all of the TX pins together and all of RX pins together - Arduinos ones.
You can do it underneath the PCB so it will be nicely at the top.

Only the horizontal yellow wires - according to photo - are required.

If you have 2 PCBs connected together - connect all of the pins from both PCBs.

1. RX and TX lines

Step 2: Connect Arduino Nodes to Raspberry Pi

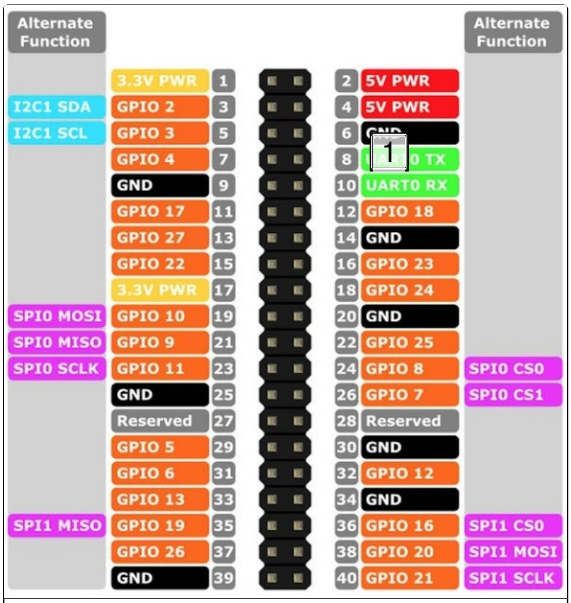
The common Arduinos RX line has to be connected to Raspberry's TX pin (GPIO 8).
Arduinos TX pins have to be connected to Raspberry's RX pin (GPIO 10) via logic level converter or with voltage divider. It is caused by difference of voltage tolerance of the Arduino and Raspberry. Arduino operates at 5V logic level and Raspberry at 3.3V. Converting the logic level is only required on the line where Ardunos TXes are connected to Raspberry's RXes, cause Arduino transmits at 5V.
On the other line the Pi is trasmitting and Arduinos are receiving so it is save without voltage divider or IIc.

Besides that RST (reset) pin of every Arduino has to be connected to Raspberry's GPIO pins. It can be done according to default pins assignment - table on the attached photo.

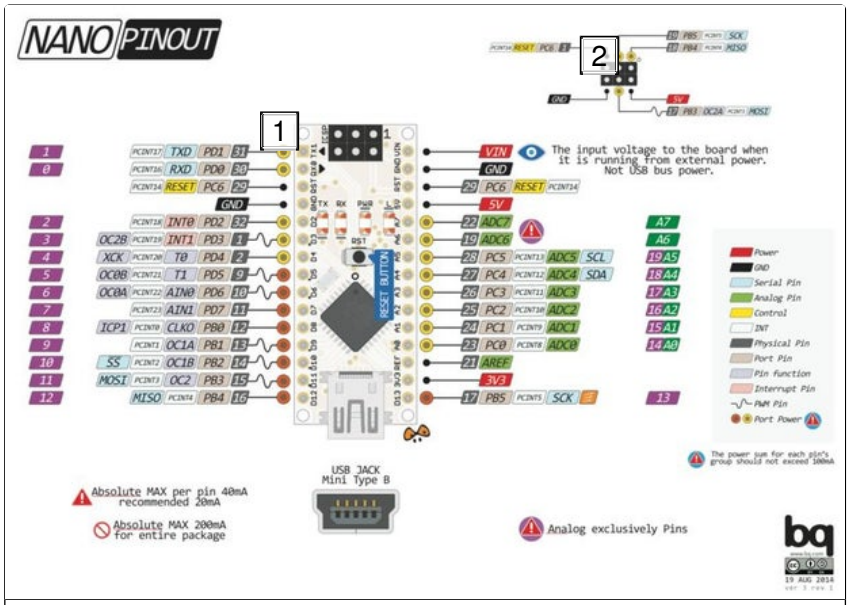
I solderer two male gold-pins to one of the Arduinos. You can do it the same way. Next I placed small PCB with the voltage divider on those pins. You don't have to do it this way. I just didn't want to have more cables than needed in my timer.

If you want to use another pins - make changes in the update.py file. You can use this command:

nano ~/RH-ota/update.py

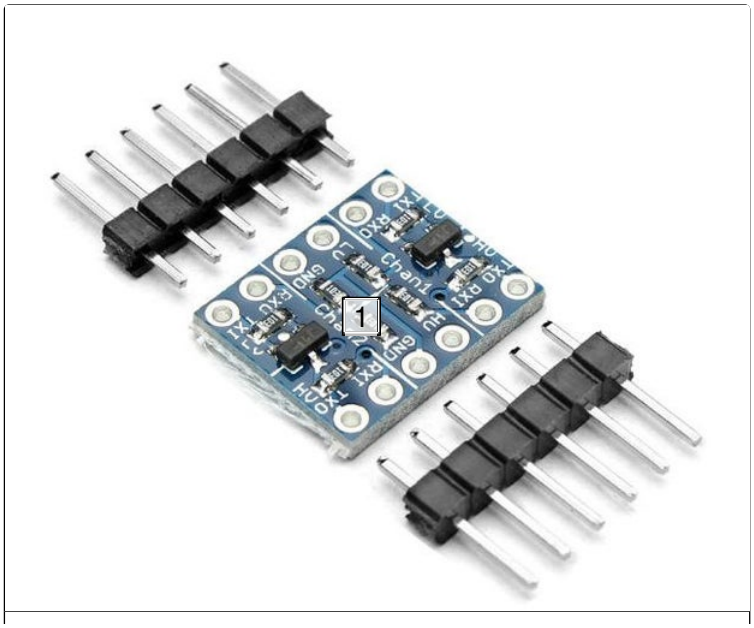


1. UART pins

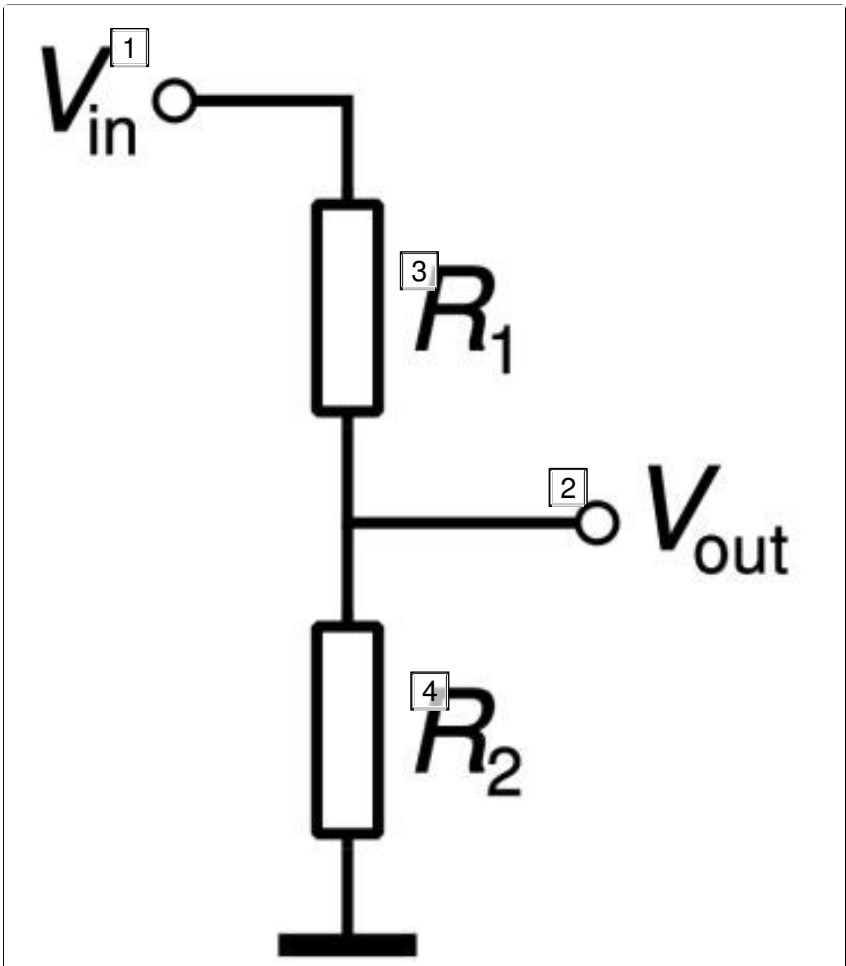


1. UART pins
2. reset pin is doubled on the ISP header - you can use that one if you want

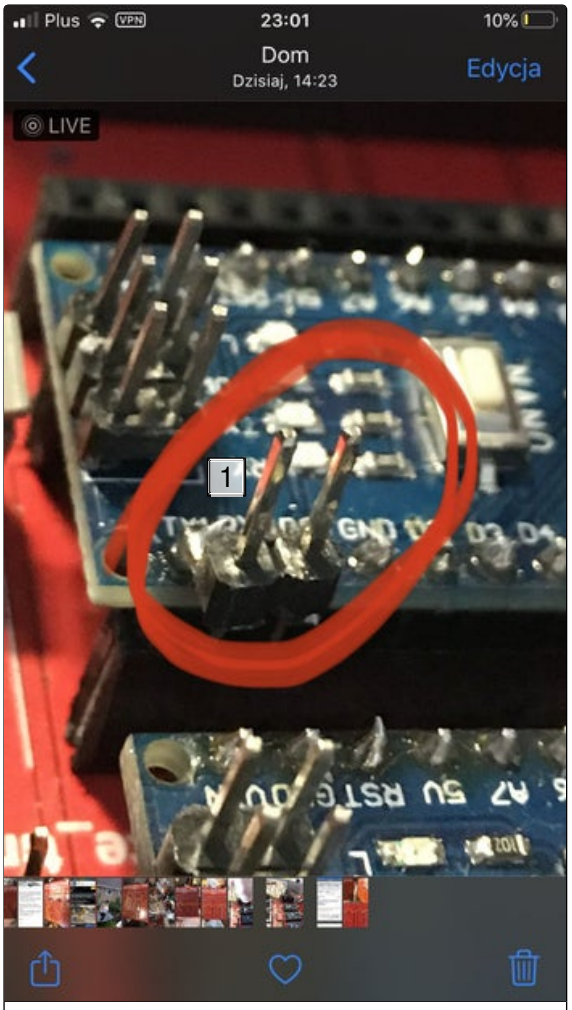
Arduino	Raspberry's GPIO pin
1	12
2	16
3	20
4	22
5	6
6	13
7	19
8	26



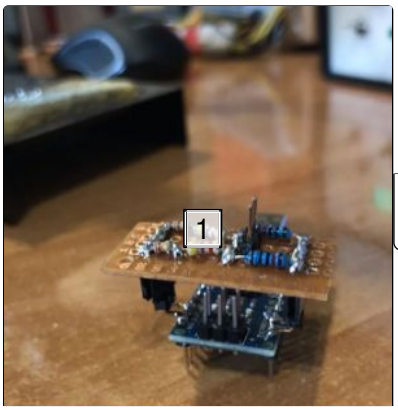
1. logic level converter



1. Arduino - TX - 5V
2. Raspberry - RX - 3.3V
3. 5kOhm
4. 10kOhm



1. additional gold-pins



1. voltage divider

Step 3: Login Into Raspberry Via SSH and Downloading the Updater

Open the ssh connection with your Raspberry. Establish connection to the internet.

You can also hook up display to the Raspberry and login into Raspbian at the Raspberry itself.

After logging into Pi download repository from github page using those commands:

git clone https://github.com/szafranski/RH-ota.git

Enter downloaded folder:

cd RH-ota

And open update script:

python update.py

If you got and error after entering first command you probably have to install git from apt.

Use command: sudo apt install git

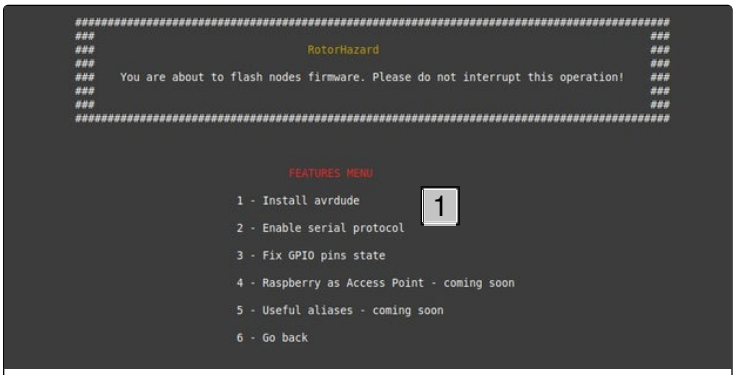
Step 4: Prepare Raspbian OS

If you are doing this for the first time you have to install software that connects with Arduino and has ability to program it. Do it by entering Additional Features menu and select point 1 - "Install avrdude".

Next you have to enable serial port on GPIO header (UART protocol) and prepare it to be connected with external device. It is utilized to be the console output by default so it is basically useless.

Enter Additional Features menu and point 2 - "Enable serial protocol".

Next you will be asked to reboot the Raspberry. Do it.



1. use first 2 after first opening

Step 5: Use Downloaded Software

After rebooting use downloaded software.

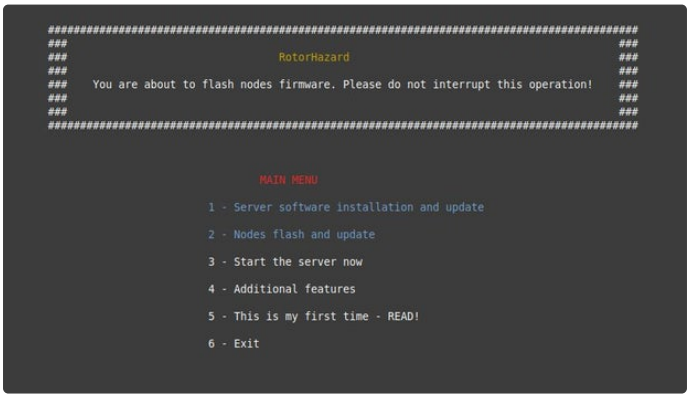
python RH-ota/update.py

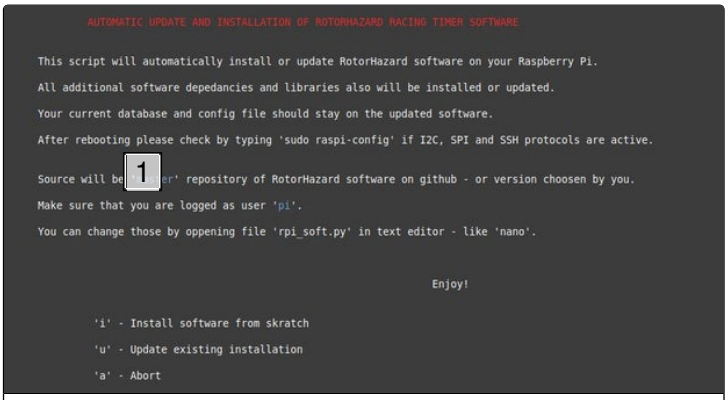
If you want to update or install (or downgrade) server software enter point 1.

If you want to flash firmware on Arduinos enter point 2.

Follow the instructions on the screen.

It should works automagically :)





1. can be changed

Step 6: Things Worth Mentioning / Troubleshoothing

Solutions to possible problems:

- Arduinos can't be flashed - check the wiring. Remember that EVERY Arduino has to be connected via reset pin and UART (RX/TX pins). All Arduinos are being reset when each of them is being flashed. It is caused by the communication on the UART line which is normally being performed if node is active. So they have to be in reset state for so of them can be flashed - software do it automatically. If even one of them is active, none of them can be programmed.
- Updating script doesn't want to open - check if python is installed and install it by using command: sudo apt install python
- Updating server software takes long time - when server is being updated, the Raspberry itself is being updated as well. If you haven't update Raspbian (Raspberry's OS) for some time it can take up to 20 minutes. It you get only errors - check internet connection.
- Few nodes updated with no problem but few of them were being flashed very slowly and I can't see them after opening the server - use option "Flash each node individually" and flash problematic nodes this way

Step 7: Things I Had to Gone Through / History of Development