



RotorHazard OTA Updater



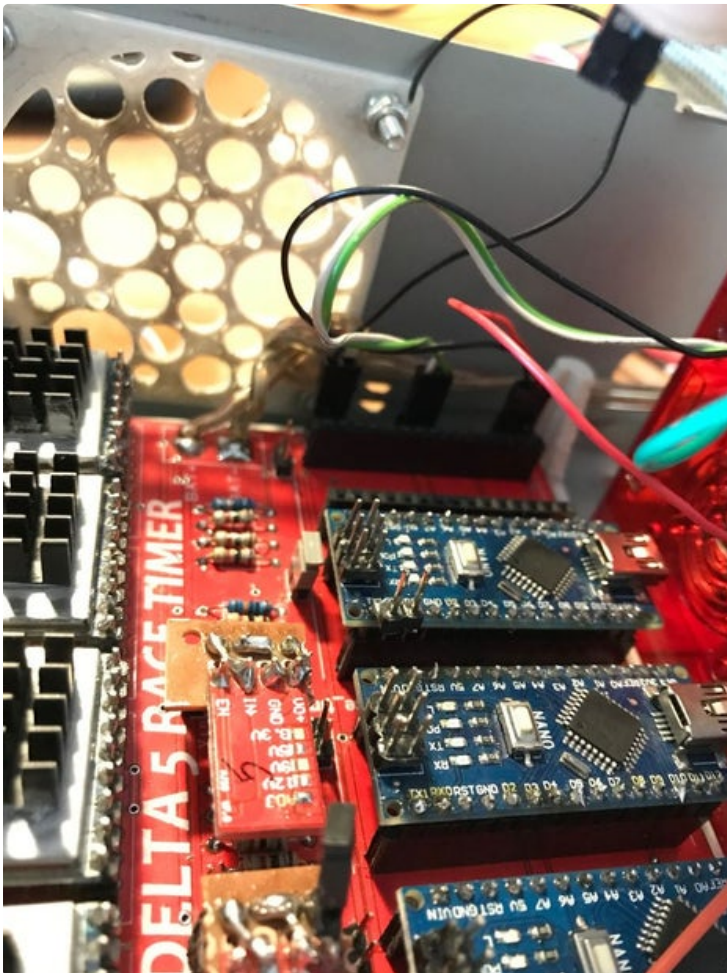
by szafranski

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
- 2. ground auto-numbering mod

Step 2: Connect Arduino Nodes to Raspberry Pi

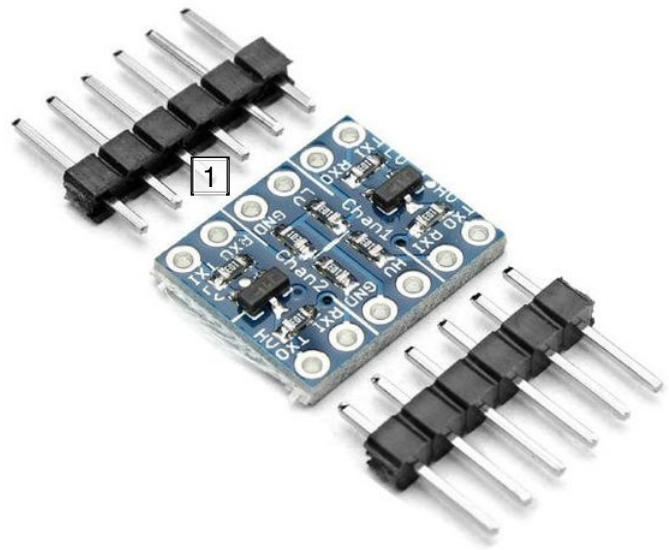
On the other line the Pi is transmitting and Arduinos are receiving so it is safe without voltage divider or IIC.

```
nano ~/RH-ota/update.py
```

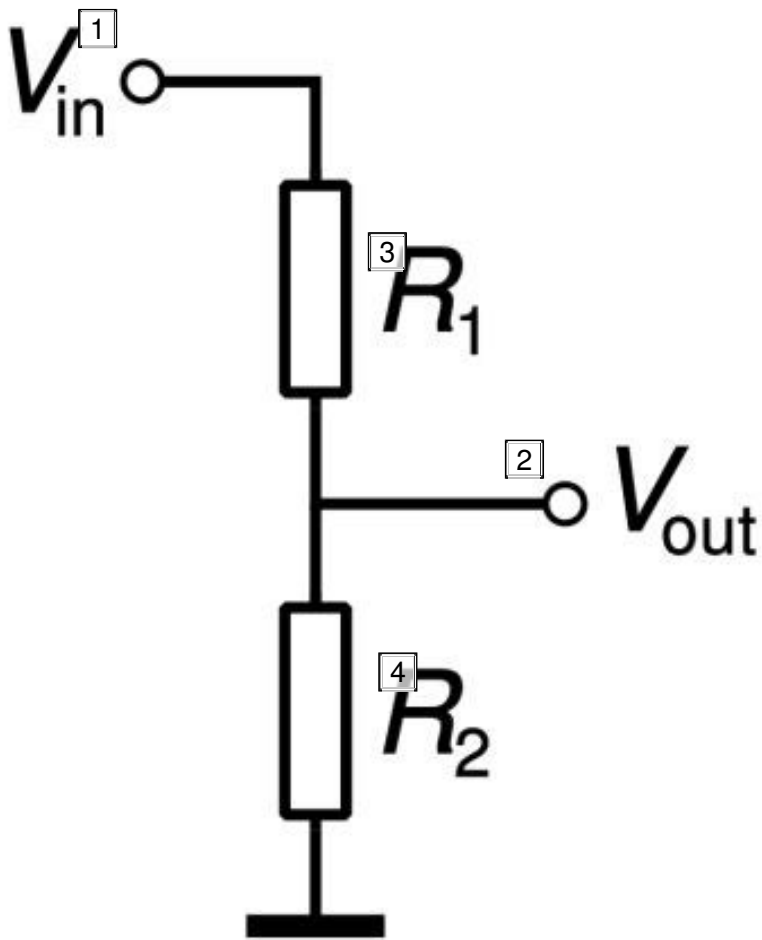
1. UART pins
2. pins used as reset pins by default



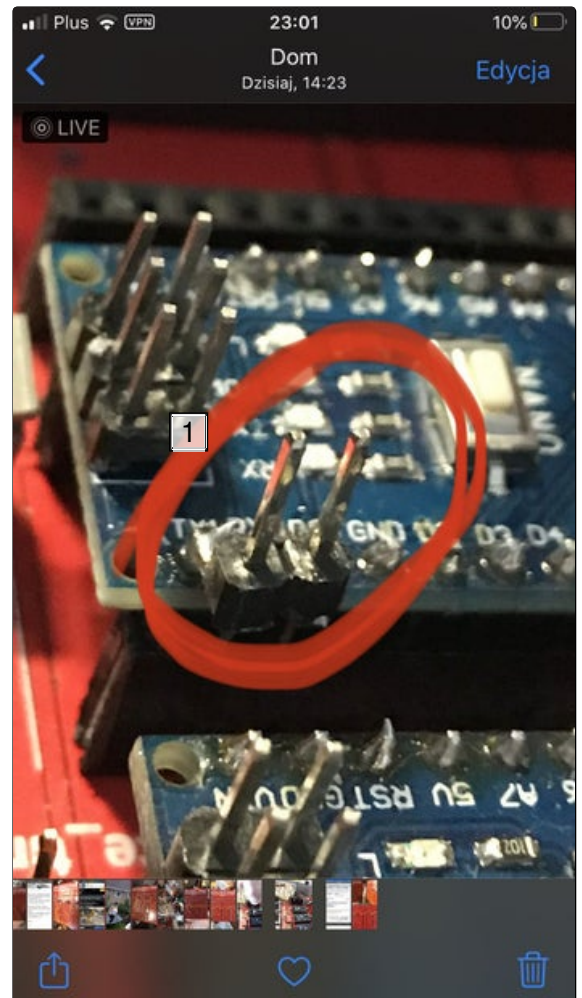
Arduino	Raspberry's GPIO pin
1	12
2	16
3	20
4	22
5	6
6	13
7	19
8	26



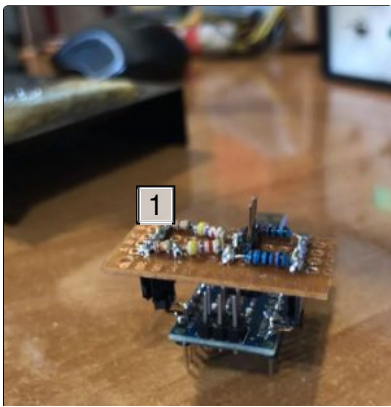
1. logic level converter



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



1. additional gold-pins on UART



1. voltage divider PCB

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 an 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.

```
#####
###                                     ###
###      RotorHazard                  ###
###                                     ###
### You are about to flash nodes firmware. Please do not interrupt this operation! ###
###                                     ###
#####

FEATURES MENU
1 Install avrdude
2 - Enable serial protocol
3 - Fix GPIO pins state
4 - Raspberry as Access Point - coming soon
5 - Useful aliases - coming soon
6 - Go back
```

1. enter '1' and '2' before using rest of the software

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 :)

```
#####
###                                     ###
###      RotorHazard                  ###
###                                     ###
### You are about to flash nodes firmware. Please do not interrupt this operation! ###
###                                     ###
#####

MAIN MENU
1 - Server software installation and update
2 - Nodes flash and update
3 - Start the server now
4 - Additional features
5 - This is my first time - READ!
6 - Exit
```

```
AUTOMATIC UPDATE AND INSTALLATION OF ROTORHAZARD RACING TIMER SOFTWARE

This script will automatically install or update RotorHazard software on your Raspberry Pi.
All additional software depedancies and libraries also will be installed or updated.
Your current database and config file should stay on the updated software.
After rebooting please check by typing 'sudo raspi-config' if I2C, SPI and SSH protocols are active.

Source will be 1 repository of RotorHazard software on github - or version choosen by you.
Make sure that you are logged as user 2.
You can change those by oppening file 'rpi_soft.py' in text editor - like 'nano'.

Enjoy!

'i' - Install software from skcratch
'u' - Update existing installation
'a' - Abort
```

1. can be changed
2. should match username of the Raspberry

Step 6: Things Worth Mentioning / Troubleshooting

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. If 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
- I tried few times and I really think that my Raspberry's UART isn't working properly - open configuration assistant on your Pi by typing 'sudo raspi-config' -> Interfacing options -> Serial and after hitting enter choose "no" for first question and "yes" to second one. Reboot if asked. You can also type 'sudo cd /dev' and then 'ls -l' and look for ttyS0. It should look as on attached image

Things Worth Mentioning / Troubleshooting

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

SPI?

All reset

Bad fuses

Lot of wiring

Will be added later