AtlasScientific™
Environmental Robotics

How to install AtlasIoTCore

Prepare the Raspberry Pi

Requirements

- Raspberry Pi 3b or later
- 16 Gb Micro SD card

Download and Install Raspberry Pi OS (AKA Raspian)

Download an install the Raspberry Pi OS (32 bit) onto an SD Card per the instructions here:

https://www.raspberrypi.org/documentation/installation/installing-images/

Install the SD card and boot your Raspberry Pi

Step 1 Download AtlasIoTCore

Download the application from the Atlas Scientific web site or by clicking HERE and unzip into a temporary directory on your computer.

Step 2 Install AtlasIOTCore

While logged in as an admin *(we'll assume Pi is still the username for this example)* create a directory for the application:

```
mkdir AtlasIoTCore
```

Using SSH or or similar, copy the contents of the directory you unzipped to the directory you just created on the RPi (~/AtlasIoTCore)

Change directory to the directory on the RPi (cd ~/AtlasIoTCore)

Change the permissions on the AtlasIOTCore application

```
chmod 777 AtlasIOTCore
```

Step 3 Run the program

To Run the program type:
./AtlasIOTCore --urls:"http:*:8080"

This will start the program, and make the user interface available on port 8080 of your Raspberry Pi

Recommendation

While the above will run the program, every time you reboot, you would have to run the program again. You probably want to install something like NGINX and setup AtlasIoTCore as a deamon

Host by Nginx with auto-restart capability

First, install and start Nginx:

```
sudo apt-get install nginx
sudo /etc/init.d/nginx start
```

Open Nginx config file:

```
sudo nano /etc/nginx/sites-available/default
```

Replace its content with:

```
server {
   listen       80 default_server;
   server_name  _;
   location / {
      proxy_pass         http://localhost:5000;
      proxy_http_version 1.1;
      proxy_set_header   Upgrade $http_upgrade;
      proxy_set_header   Connection keep-alive;
      proxy_set_header   Host $host;
      proxy_cache_bypass $http_upgrade;
      proxy_set_header   X-Forwarded-For $proxy_add_x_forwarded_for;
      proxy_set_header   X-Forwarded-Proto $scheme;
   }
}
```

server_name is set to _; means I don't limit domain names, just accept any hostnames.
proxy_pass is the default Kestrel HTTP endpoint, which runs our ASP.NET Core website.

Check and apply the config file:

```
sudo nginx -t
sudo nginx -s reload
```

Now, start your ASP.NET Core website, you will be able to access your Raspberry Pi using port 80.

```
./AtlasIOTCore
```

Now we have one last step to go, make dotnet process auto-restart in case your code accidentally blows up. To do that, we will create a systemd service.

```
sudo nano /etc/systemd/system/AtlasIoT.service
```

```
[Unit]
Description=AtlasIoT Service

[Service]
WorkingDirectory=/home/AtlasIoT
ExecStart=/home/pi/AtlasIoT
Restart=always
# Restart service after 10 seconds if the dotnet service crashes:
RestartSec=10
KillSignal=SIGINT
SyslogIdentifier=AtlasIoT
User=pi
Environment=ASPNETCORE_ENVIRONMENT=Production
Environment=DOTNET_PRINT_TELEMETRY_MESSAGE=false

[Install]
WantedBy=multi-user.target
```

Note, we can only use absolute path in systemd configuration.
Register and start the service:

```
sudo systemctl enable AtlasIoT.service
sudo systemctl start AtlasIoT.service
sudo systemctl status AtlasIoT.service
```

Now, you have the auto start and failure restart capability.
Try to reboot your Raspberry Pi, the website will auto start!