Seminár Robotika.SK

JetBot

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny

www.robotika.sk/seminar/2020/jetbot.pdf
https://github.com/andylucny/JetBotDemos.git

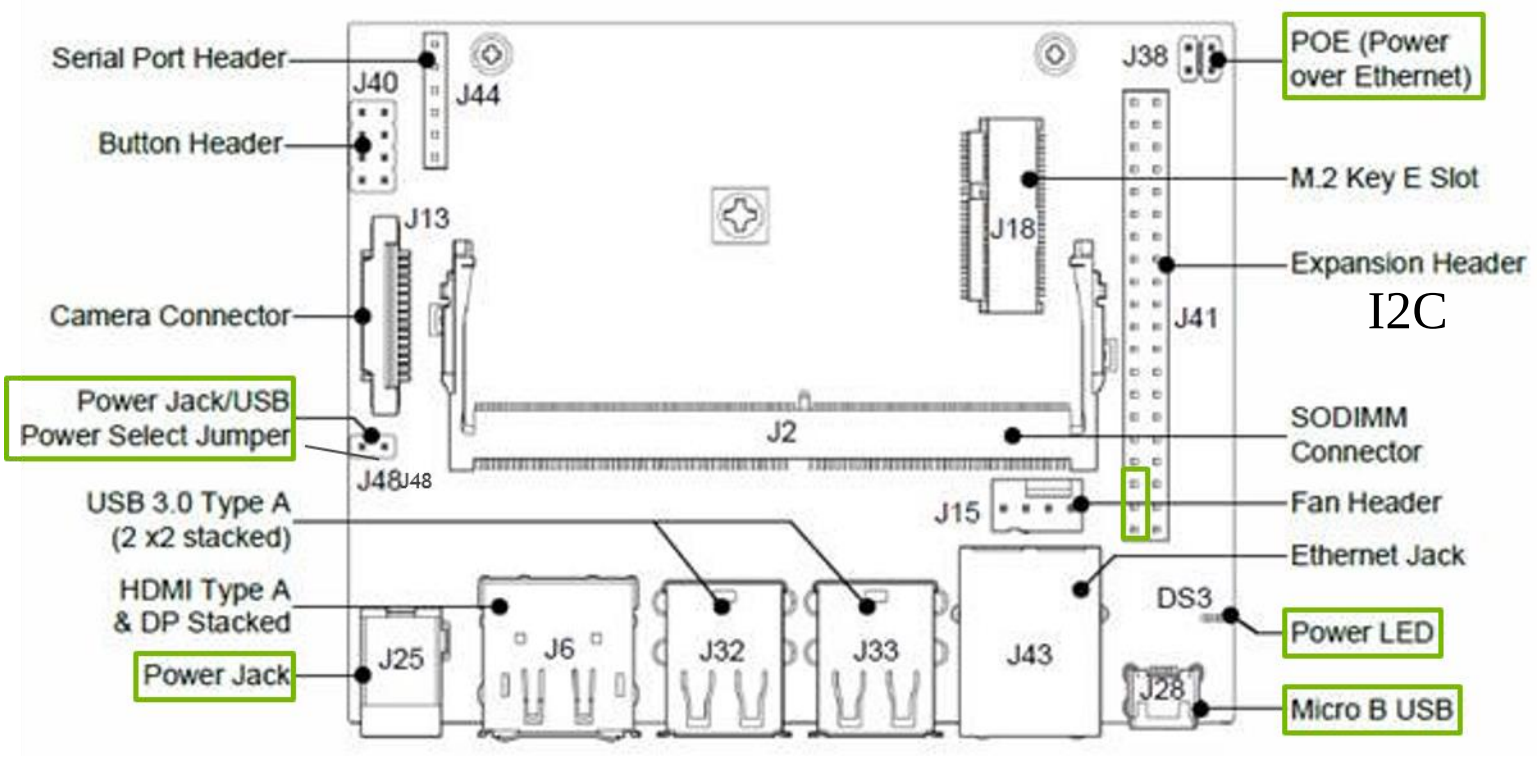
Jetson Nano Developer Kit

- embeded computer od NVIDIA
- určený pre Edge AI
- CPU: Quad-core ARM A57 @ 1.43 GHz (**arm64**)
- GPU: 128-core Maxwell
- RAM 4 GB 64-bit LPDDR4
- Display port, HDMI, 4x USB 3.0, 1G Ethernet
- (GPIO, I2C, I2S, SPI, UART)
- rozmery: 69 mm x 45 mm
- cena: 99-130 USD

Spotreba a napájanie

- Spotreba Jetson Nano je 10W
- Pri nábehu potrebuje prúd 3A
- Nie je vôbec ľahké nájsť vhodný zdroj s ktorým by sa rozbehol
- Ja som použil MW GS18E05-P1J 5-6V direct, 3.0-2.5A

Jumpre a rozhrania



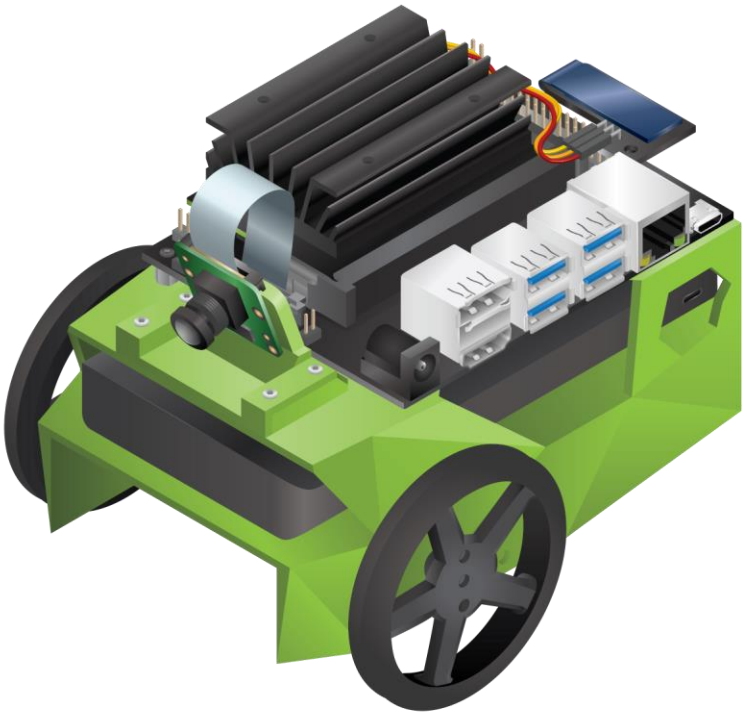
Pri napájaní cez J25 má byť zavretý jumper J48, inak sa napája cez J28

I2C

Jetson Nano J41 Header					
Sysfs GPIO	Name	Pin	Pin	Name	Sysfs GPIO
	3.3 VDC *Power*	1	2	**5.0 VDC** *Power*	
	I2C_2_SDA *I2C Bus 1*	3	4	**5.0 VDC** *Power*	
	I2C_2_SCL *I2C Bus 1*	5	6	**GND**	
gpio216	**AUDIO_MCLK**	7	8	**UART_2_TX** */dev/ttyTHS1*	
	GND	9	10	**UART_2_RX** */dev/ttyTHS1*	
gpio50	**UART_2_RTS**	11	12	**I2S_4_SCLK**	gpio79
gpio14	**SPI_2_SCK**	13	14	**GND**	
gpio194	**LCD_TE**	15	16	**SPI_2_CS1**	gpio232
	3.3 VDC *Power*	17	18	**SPI_2_CS0**	gpio15
gpio16	**SPI_1_MOSI**	19	20	**GND**	
gpio17	**SPI_1_MISO**	21	22	**SPI_2_MISO**	gpio13
gpio18	**SPI_1_SCK**	23	24	**SPI_1_CS0**	gpio19
	GND	25	26	**SPI_1_CS1**	gpio20
	I2C_1_SDA *I2C Bus 0*	27	28	**I2C_1_SCL** *I2C Bus 0*	
gpio149	**CAM_AF_EN**	29	30	**GND**	
gpio200	**GPIO_PZ0**	31	32	**LCD_BL_PWM**	gpio168
gpio38	**GPIO_PE6**	33	34	**GND**	
gpio76	**I2S_4_LRCK**	35	36	**UART_2_CTS**	gpio51
gpio12	**SPI_2_MOSI**	37	38	**I2S_4_SDIN**	gpio77
	GND	39	40	**I2S_4_SDOUT**	gpio78

https://www.jetsonhacks.com/nvidia-jetson-nano-j41-header-pinout/

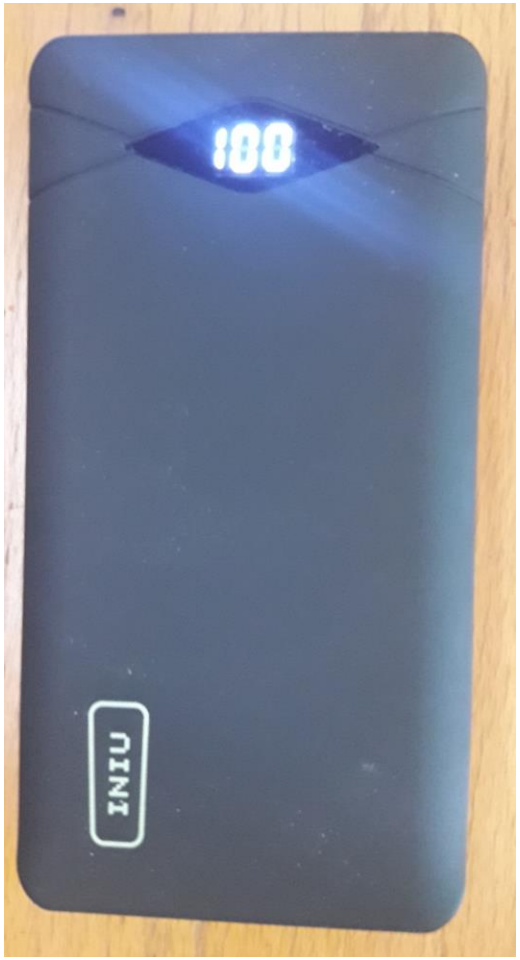
JetBot



- open source projekt od NVIDIA na báze Jetson Nano
- kamera z Raspberry Pi
- wifi
- LED display
- napájanie z powerbanky
- dva elektromotory s prevodovkou a controller
- cena cca 350 eur

https://github.com/NVIDIA-AI-IOT
https://github.com/NVIDIA-AI-IOT/jetbot/wiki
https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetbot-ai-robot-kit/

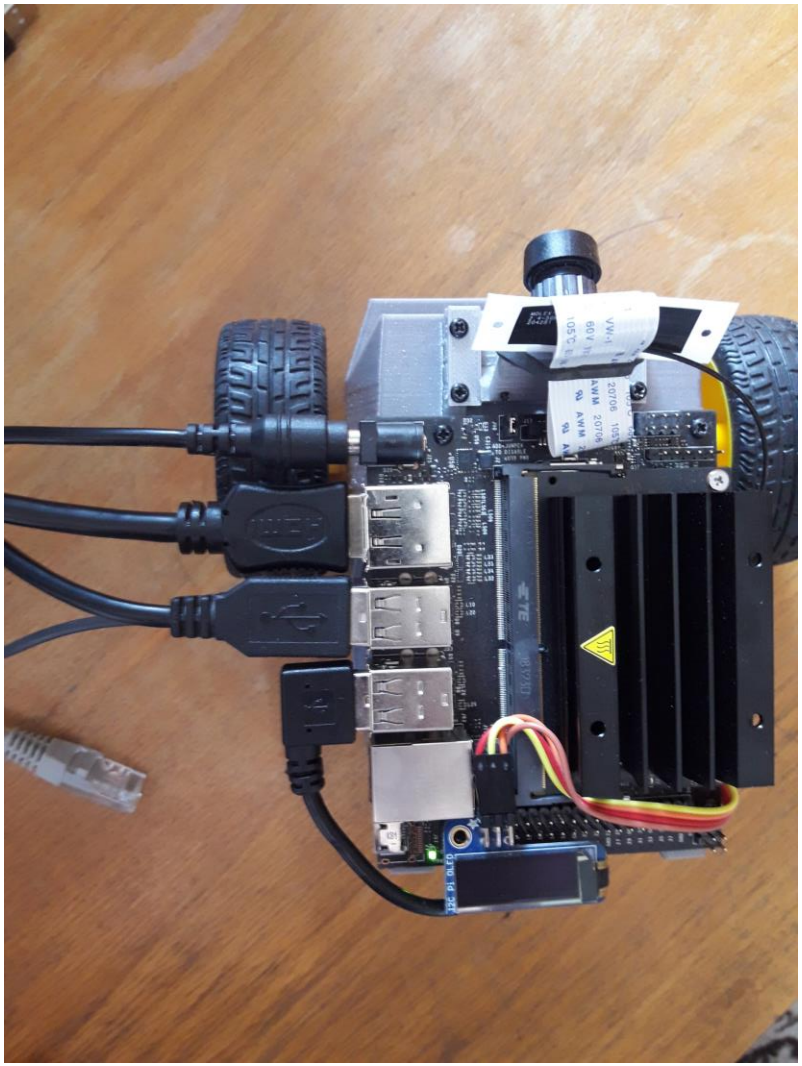
Spotreba a napájanie



- za 1½ hod. používania JetBota klesne powerbanka z 100% na 91% za 2 ½ hod. na 85%

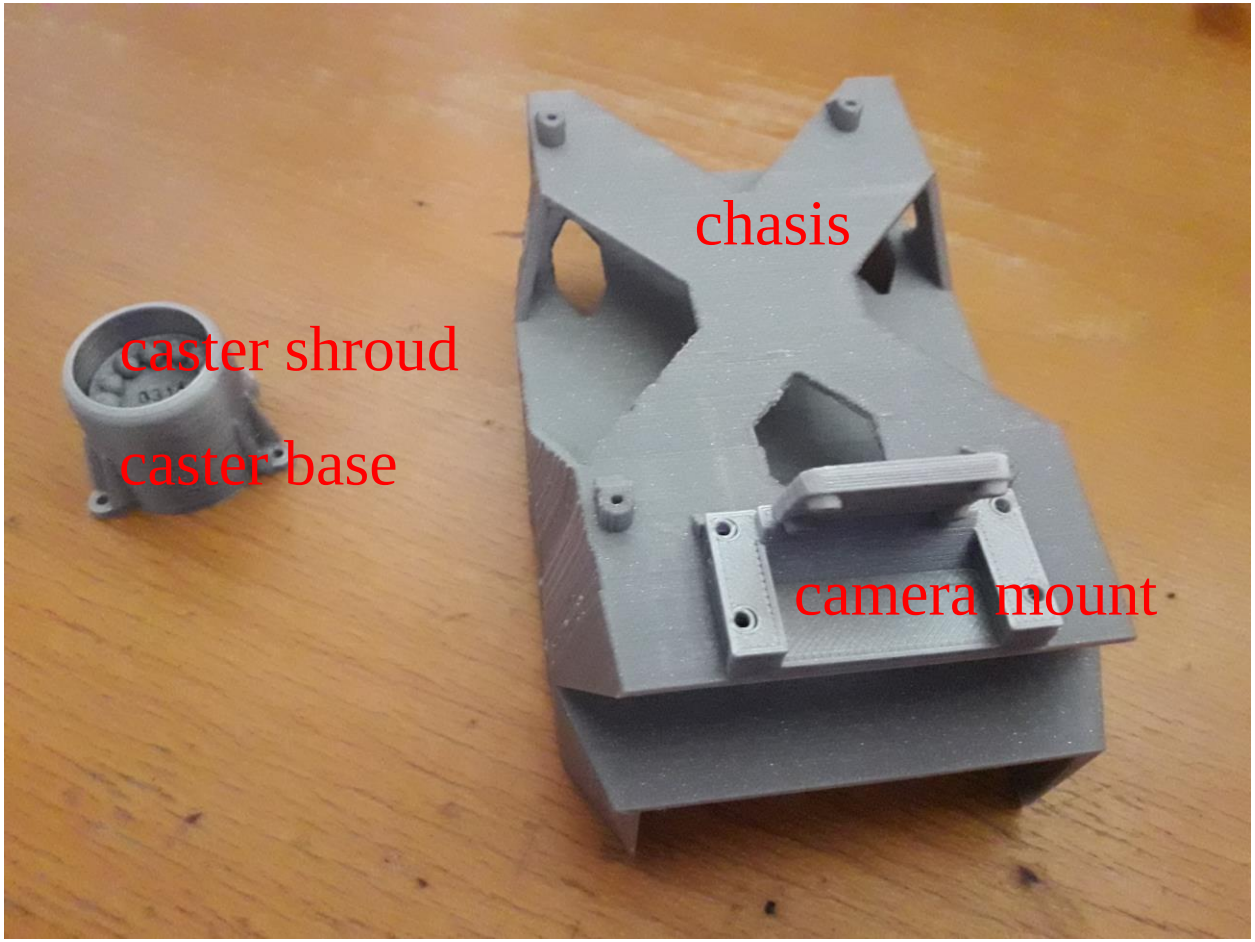
- Jetbot je napájaný z databanky pre mobil
- Nerozbehne sa však z hociktorej, len z tejto jednej (INIU)
- Nabíjačka obyčajná (USB)
- Káble špeciálne! (malý odpor)

Napájanie zo zdroja



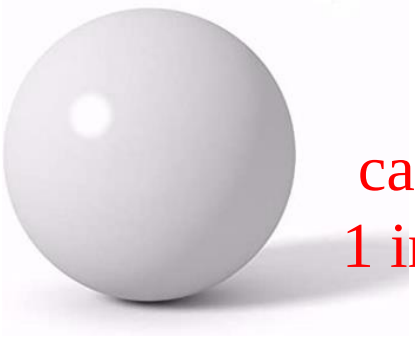
- Pri vývoji je Jetbot možné napájať aj zo zdroja
- Jumper J48 je zavretý, aby Jetson Nano bral energiu len zo zdroja
- Pritom podvozok napájame z USB

3D tlač dielov Jetbota



Jetbot pozostava zo 4 dielov, ktoré treba vytlačiť
.stl je na https://github.com/NVIDIA-AI-IOT/jetbot/tree/master/assets
.gcode pre Prusa (pre kolesa 65mm) je k dispozíci na
www.robotika.sk\seminar\2020\jetbot-prusa.zip

Dokúpená mechanika



caster ball,
1 inch

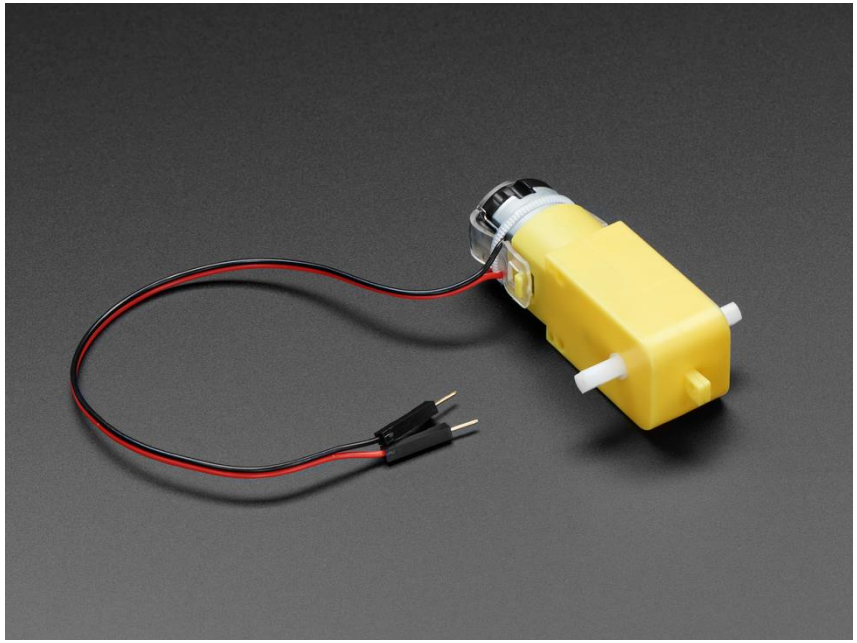
wheels,
65mm

Skrutky: samorez M2 8mm 20 ks, M3 25mm 4ks + matica

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

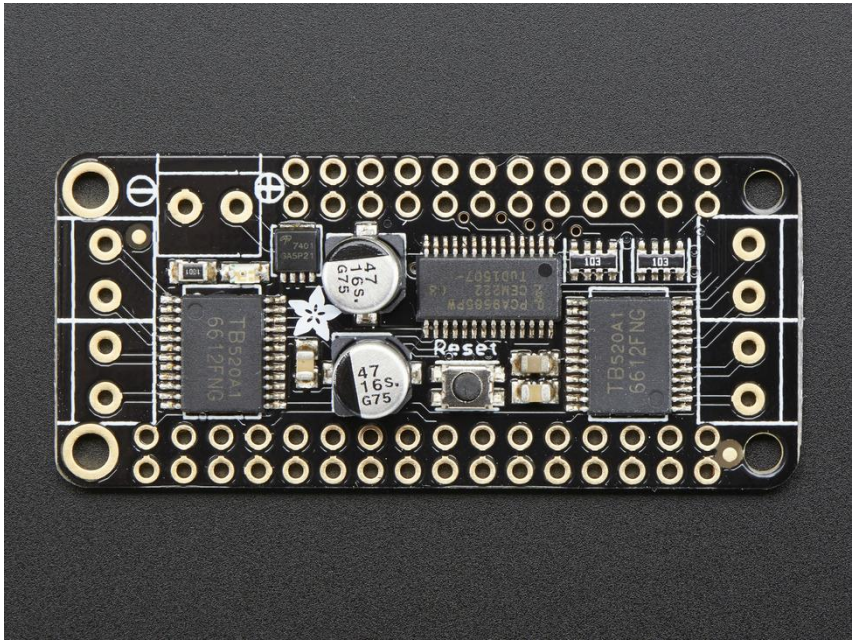
Dokúpená elektronika

2x



motor

1x



motor driver

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

Dokúpená elektronika

1x FOV 44º

kamera z Raspberry Pi (CSI)

FOV 80º 1x

objektív

10 cm
mŕtva zóna

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

Dokúpená elektronika

1x

Wifi modul
(montuje sa zospodu na dosku s procesorom)

1x

Wifi anténa

má to podporovať aj USB Edimax wifi ale je to na ľavačku i pomalšie

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

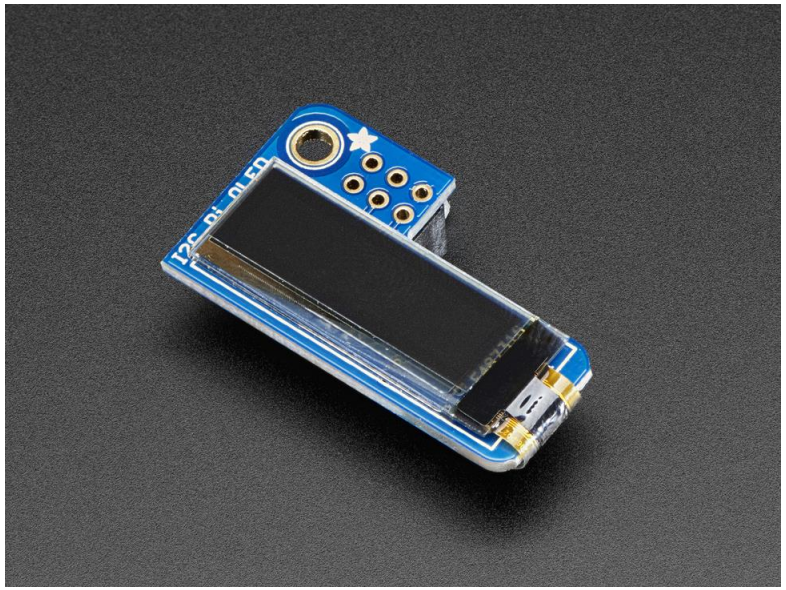
Dokúpená elektronika

2x usb kábel (1 pack)

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

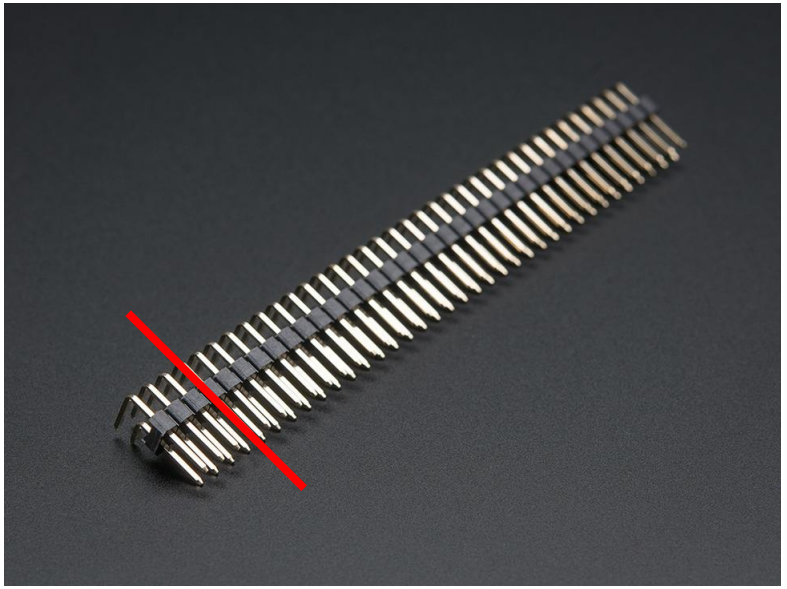
Dokúpená elektronika

1x



display

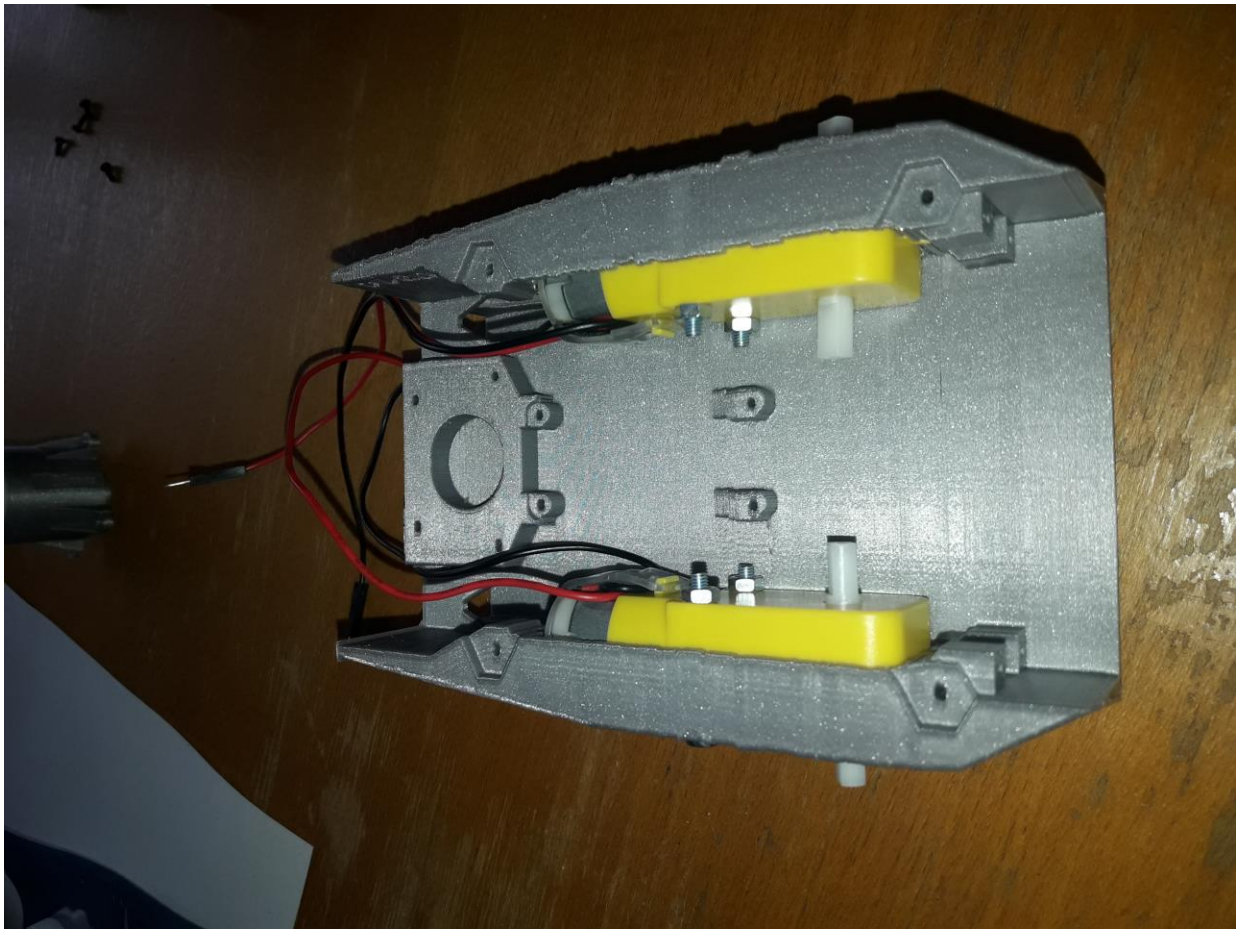
1x



pin string right angle

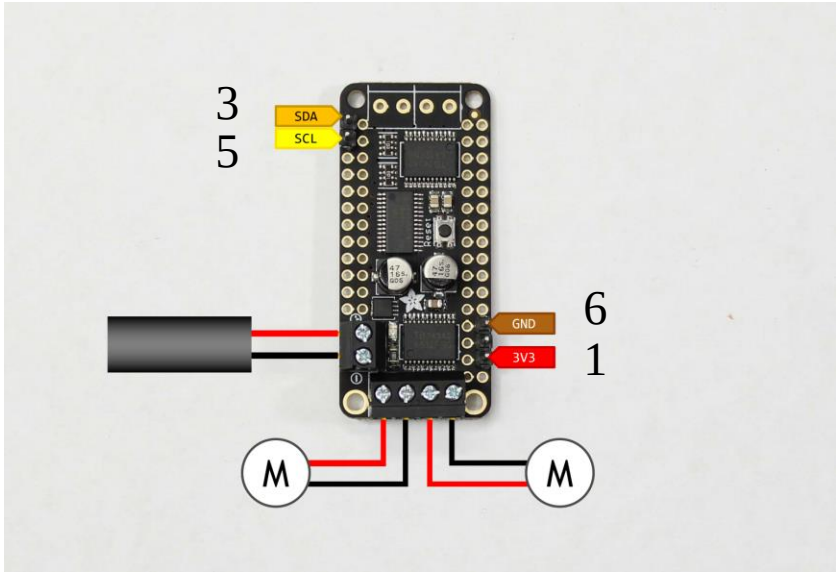
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/bill-of-materials

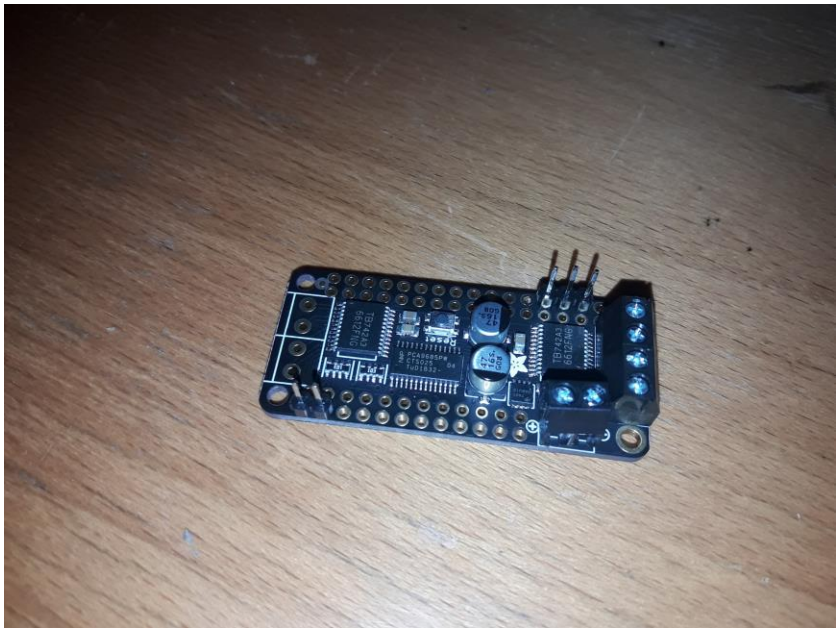
Montáž



https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

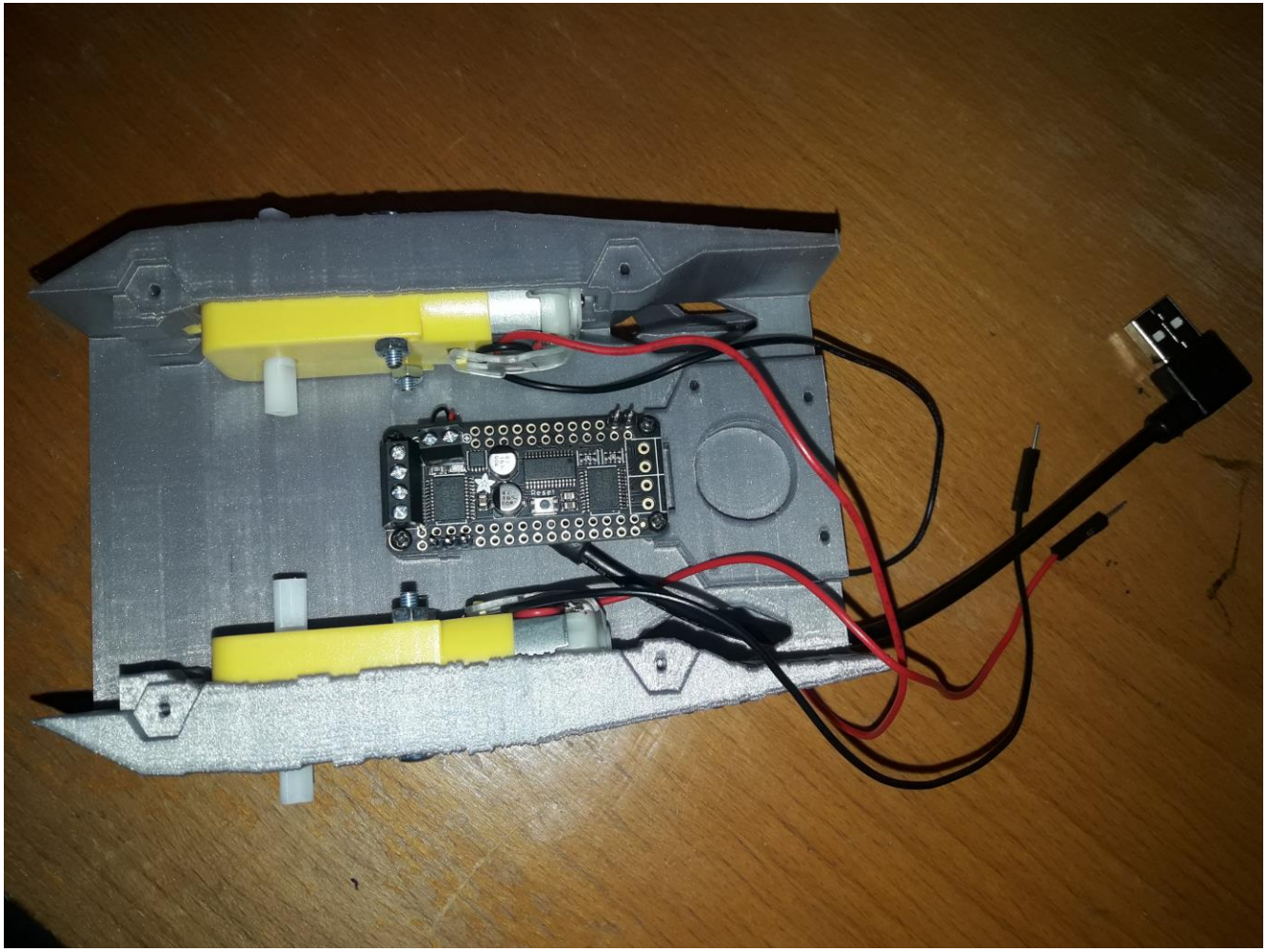
Montáž





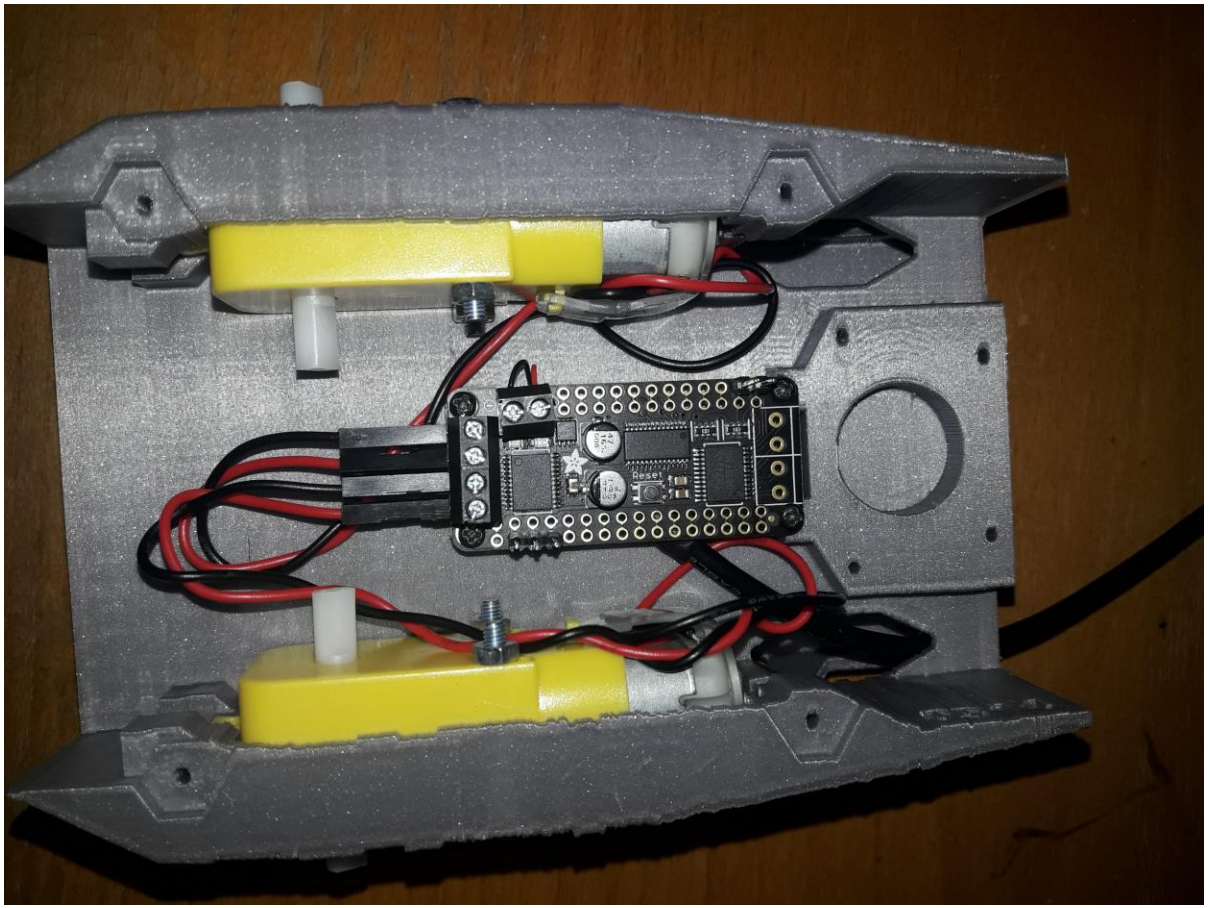
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



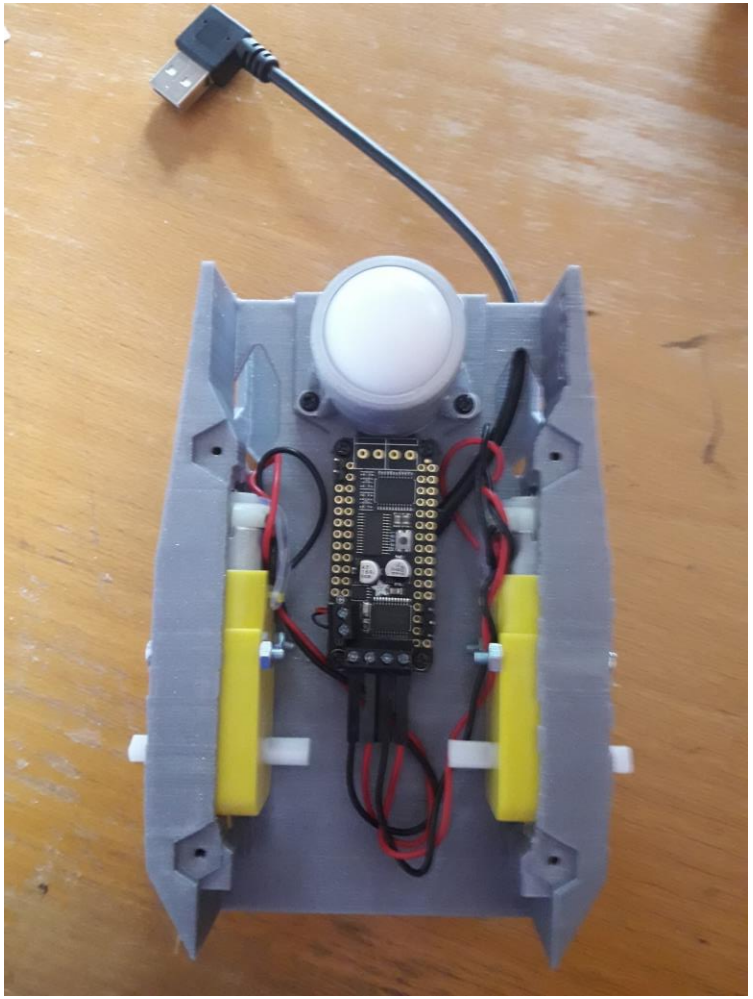
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



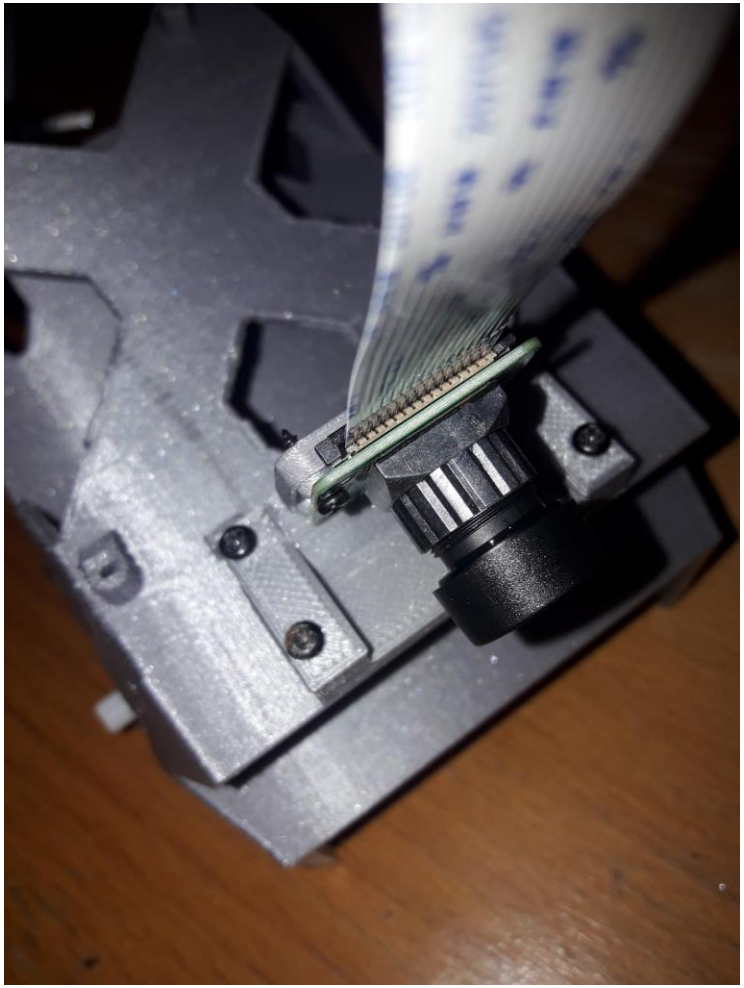
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



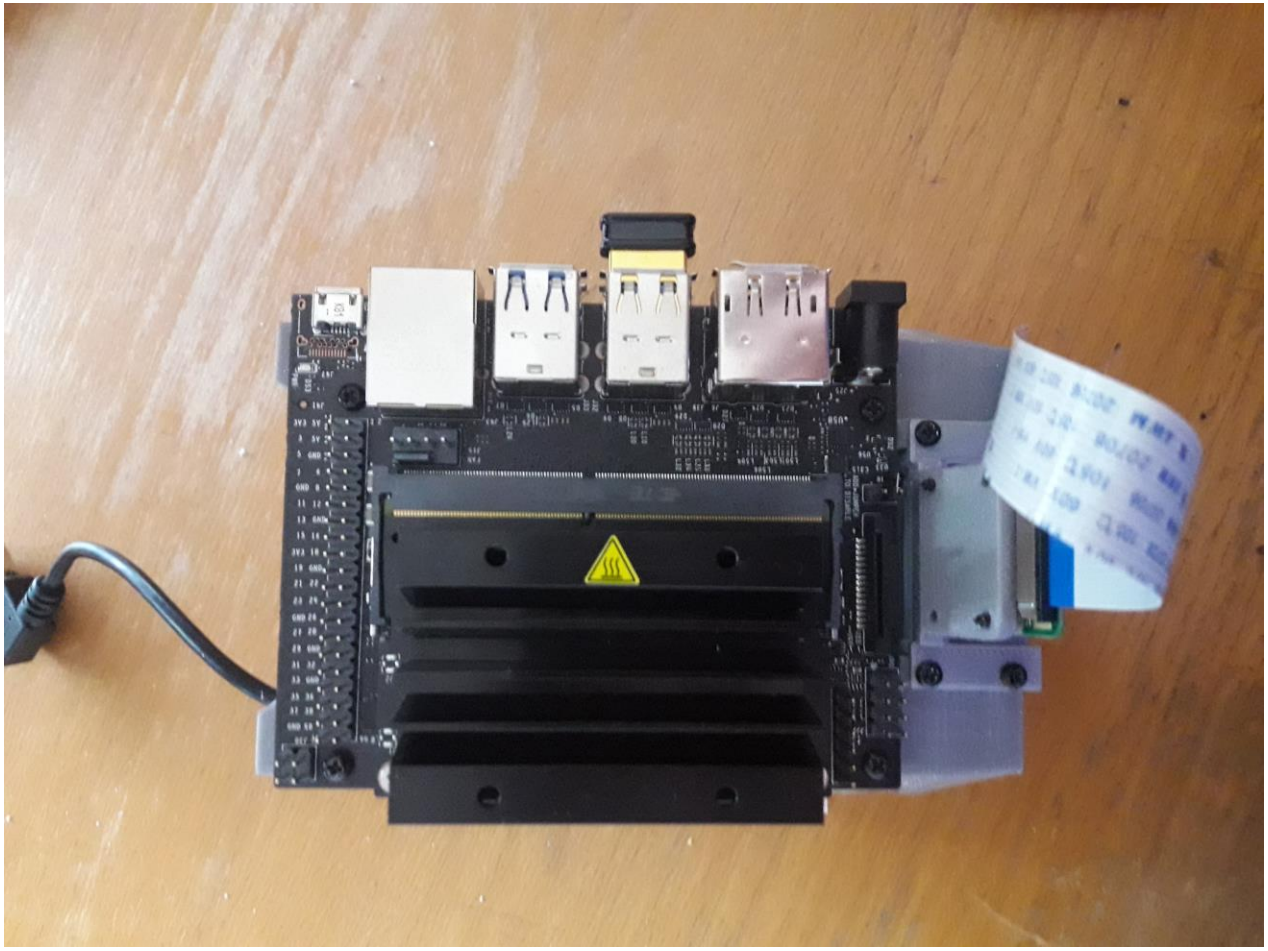
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



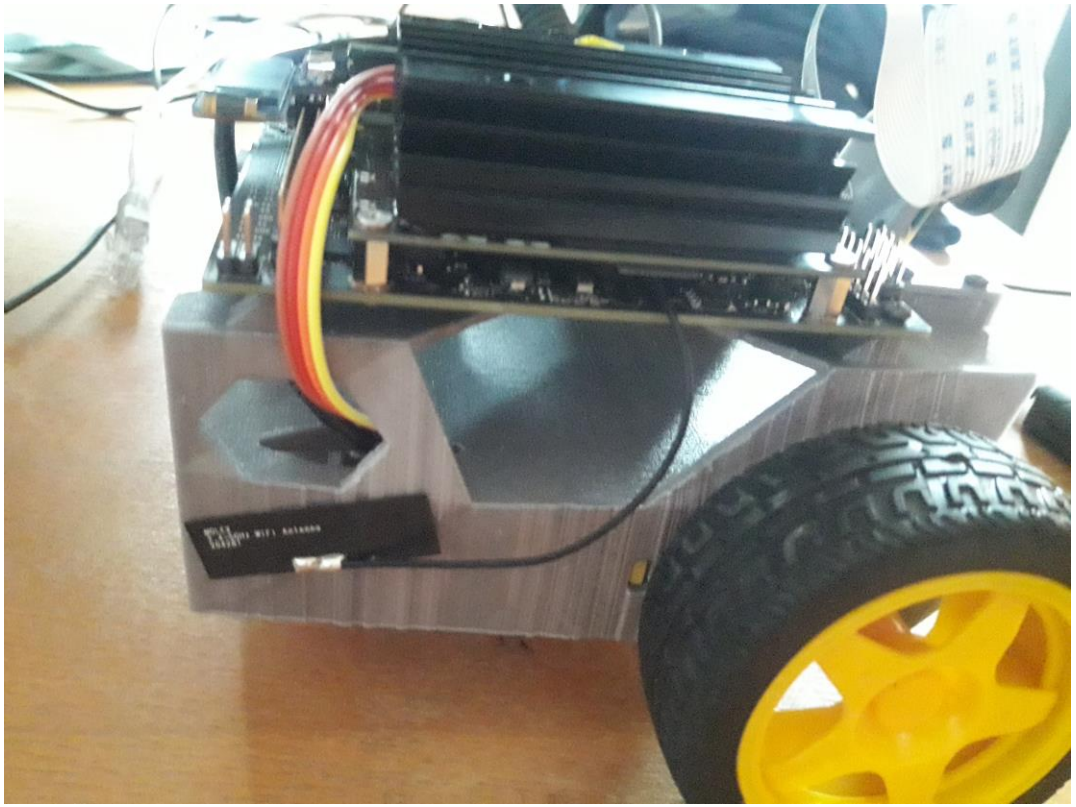
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



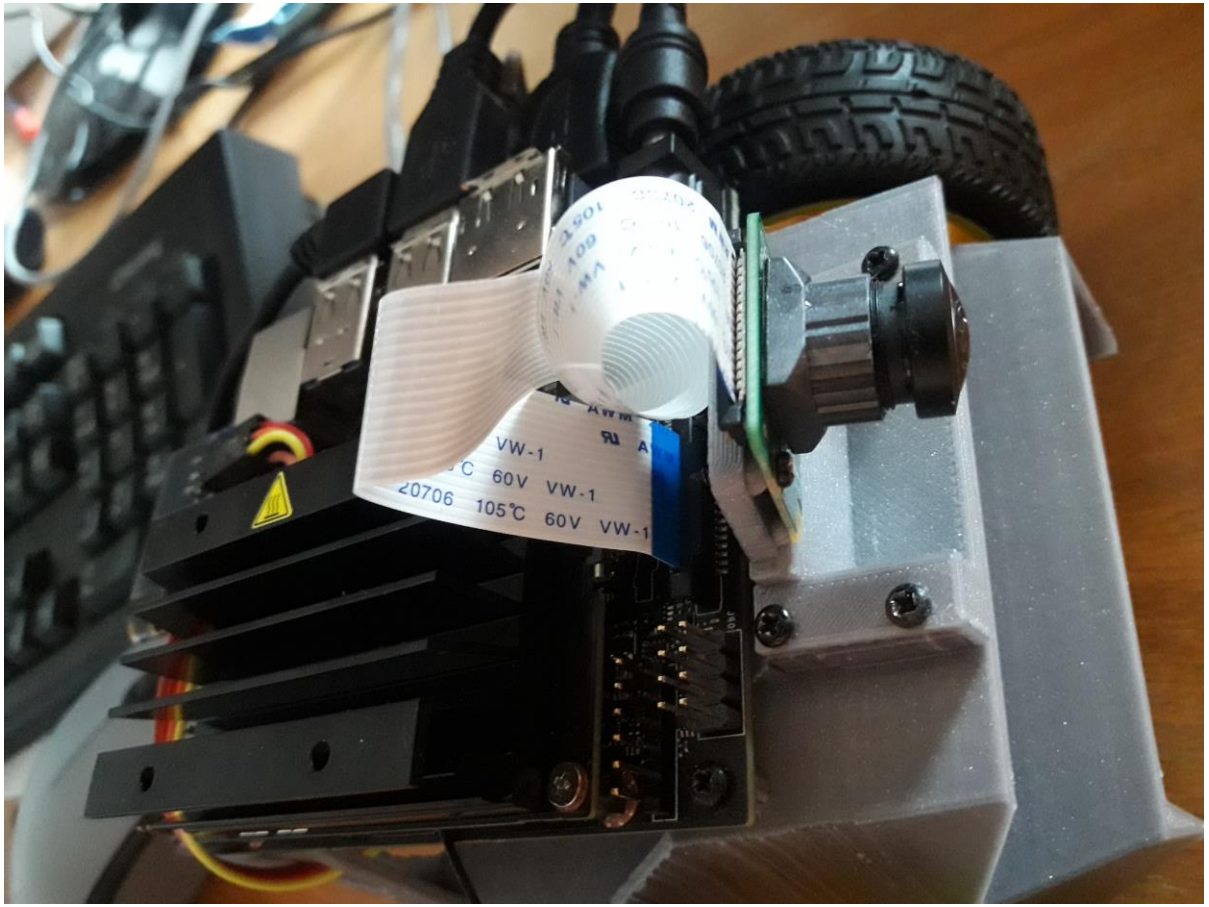
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



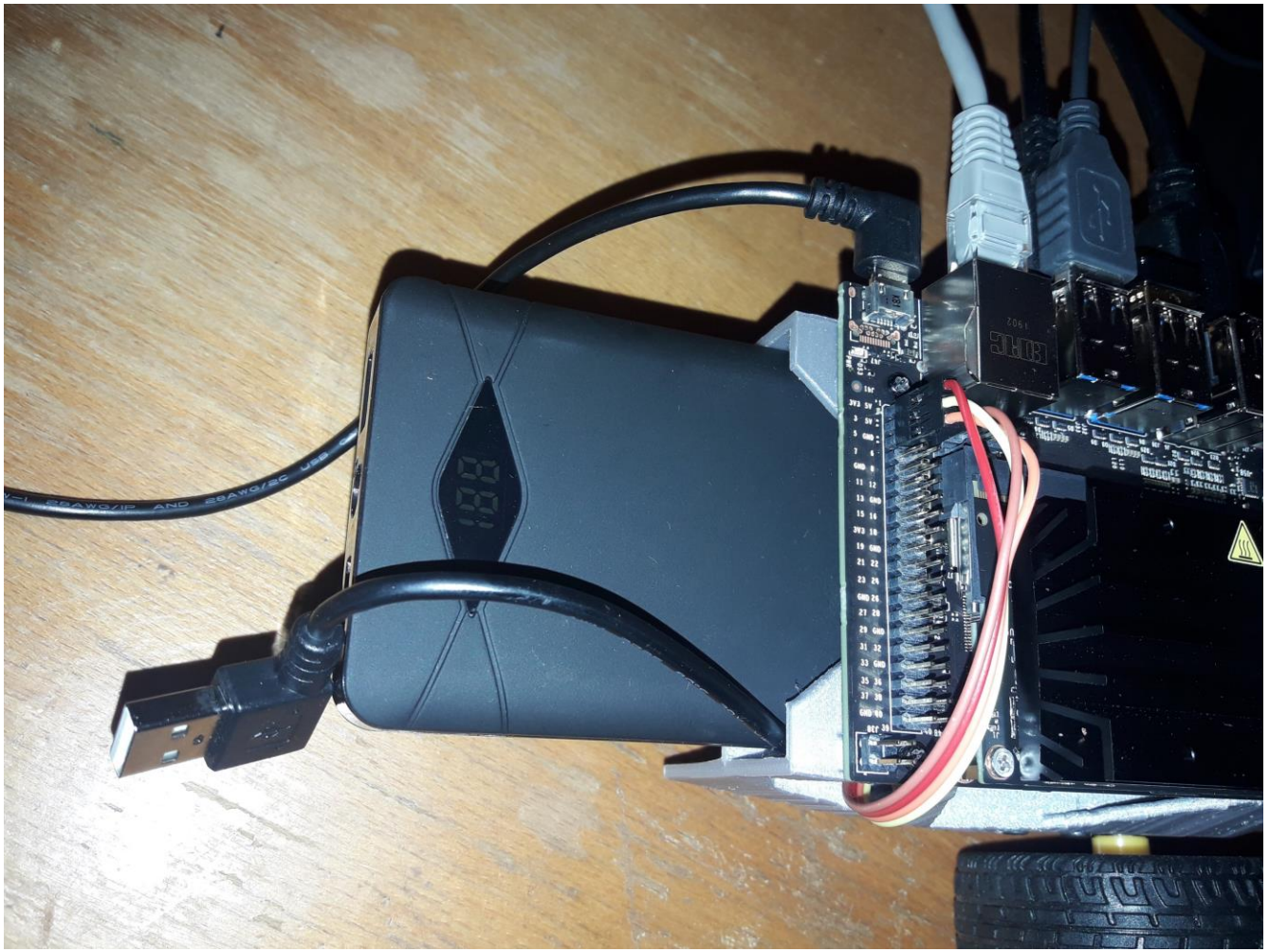
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



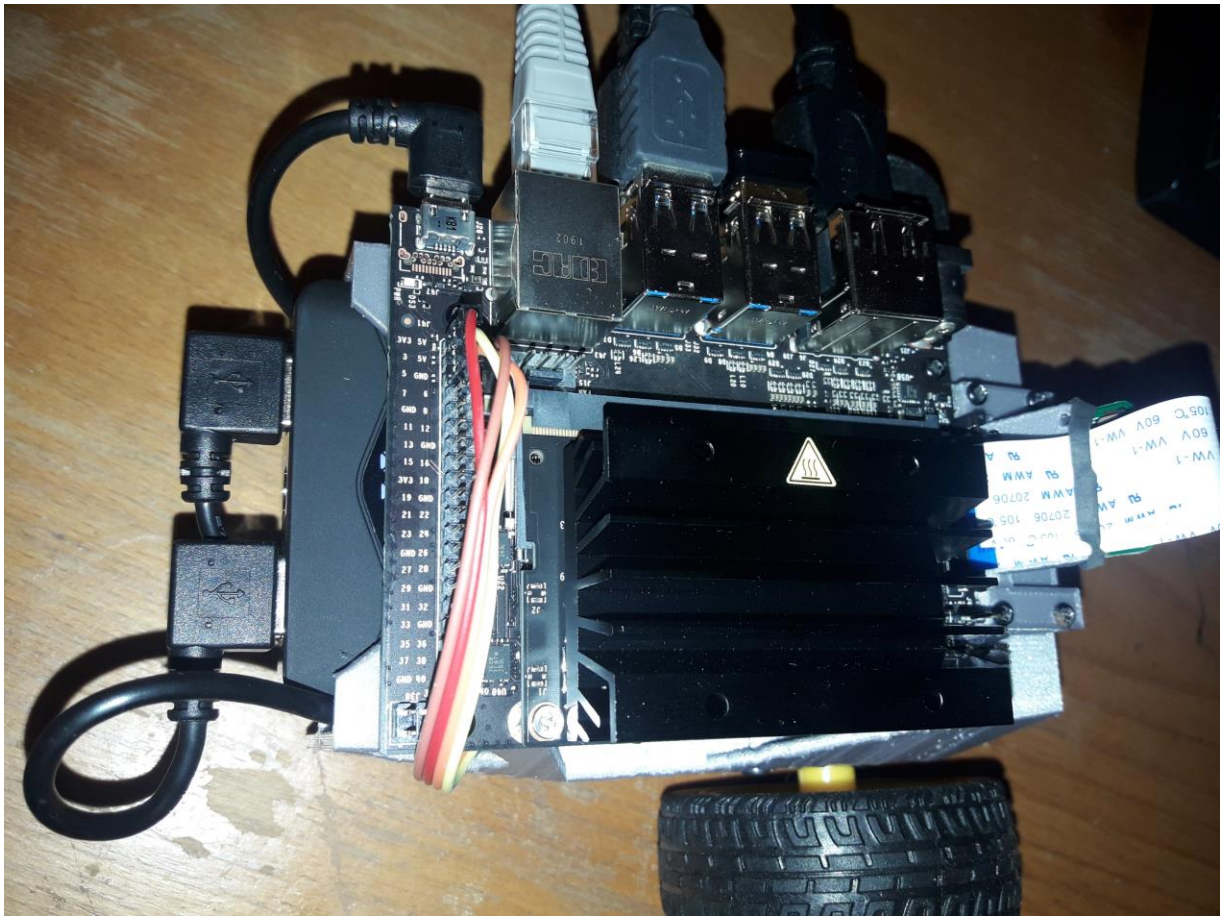
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



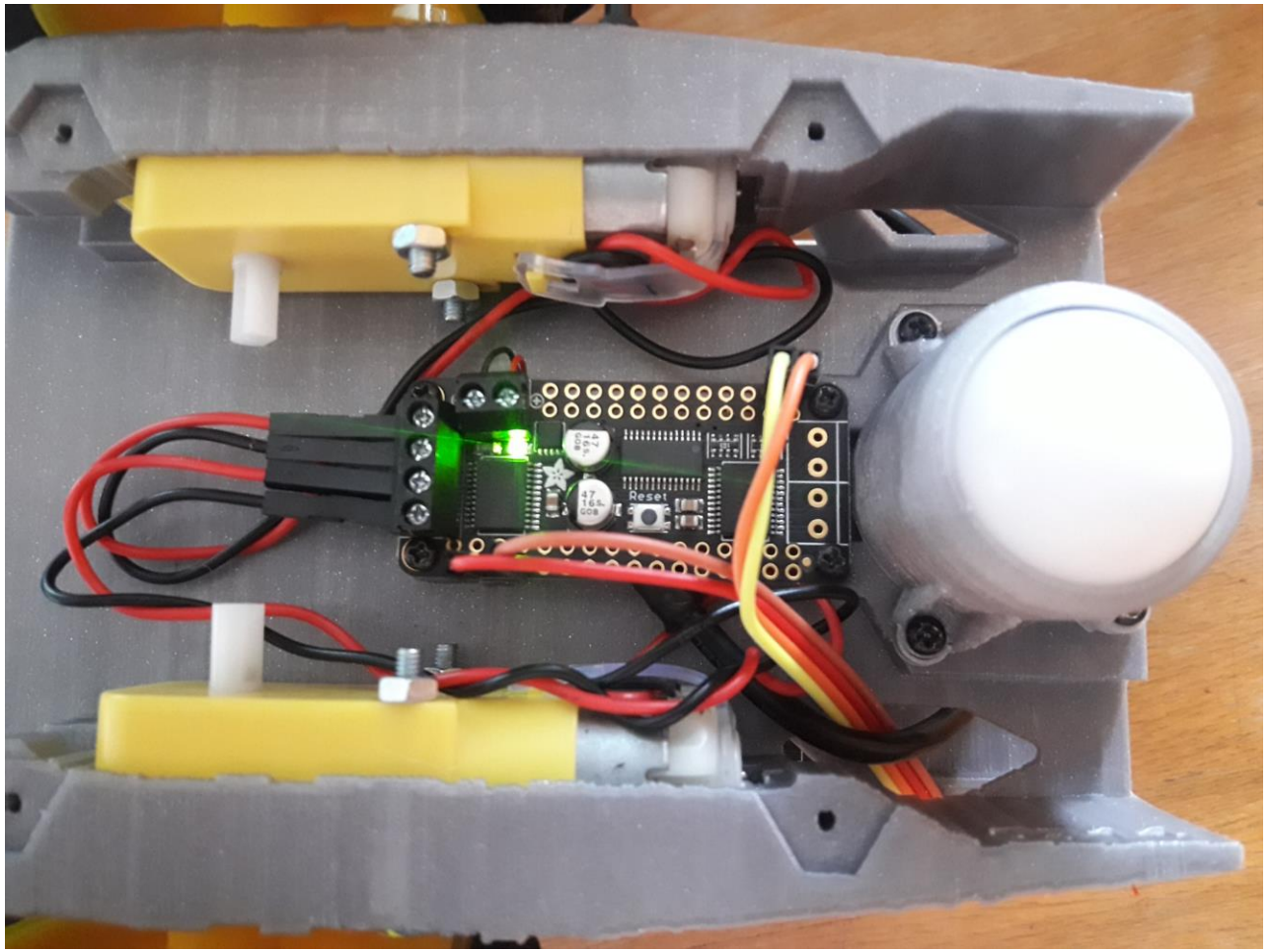
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



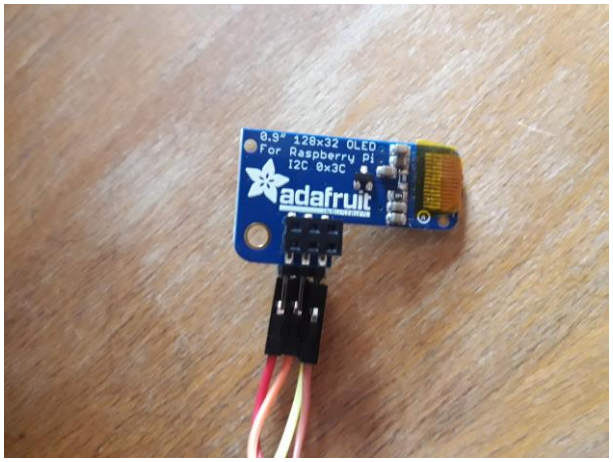
https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

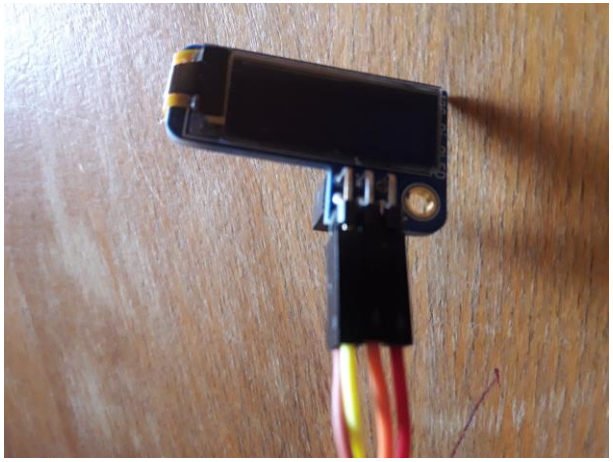
Montáž

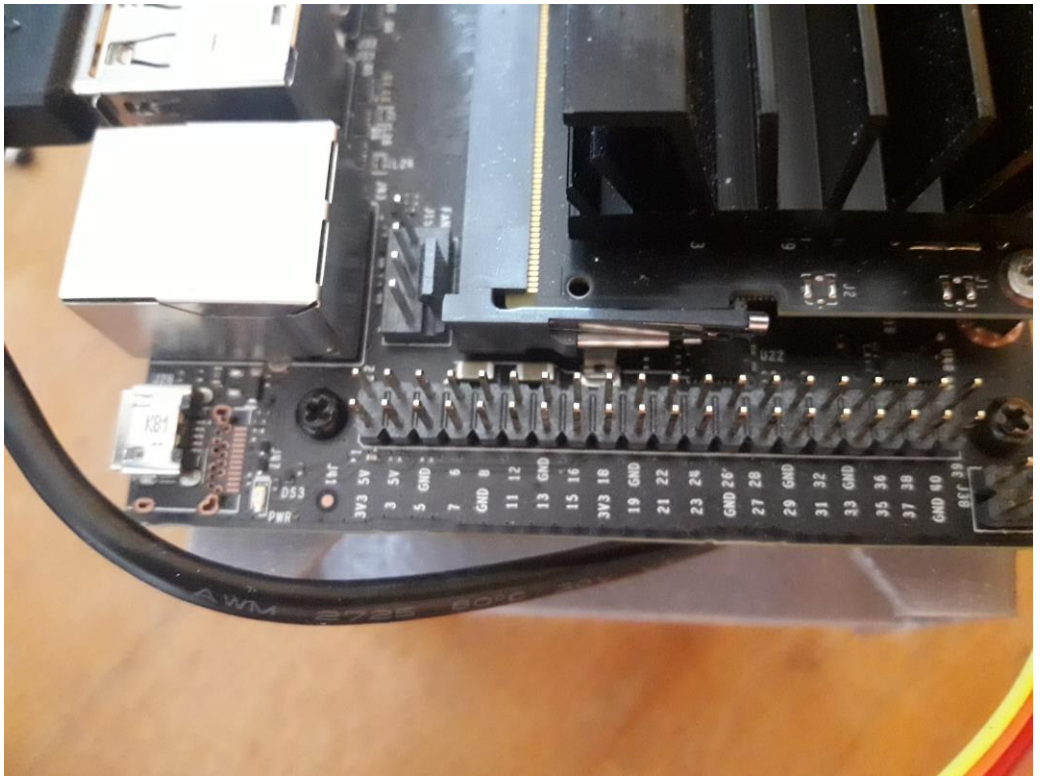


https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž

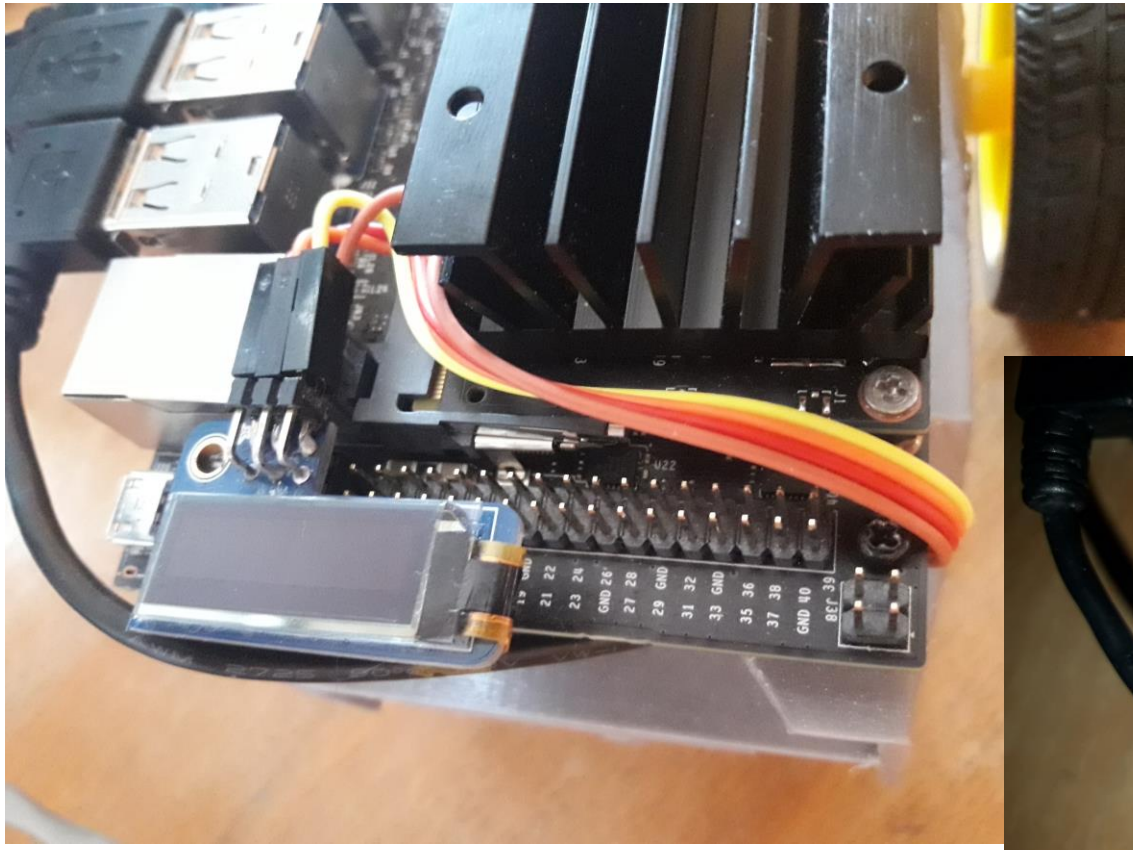


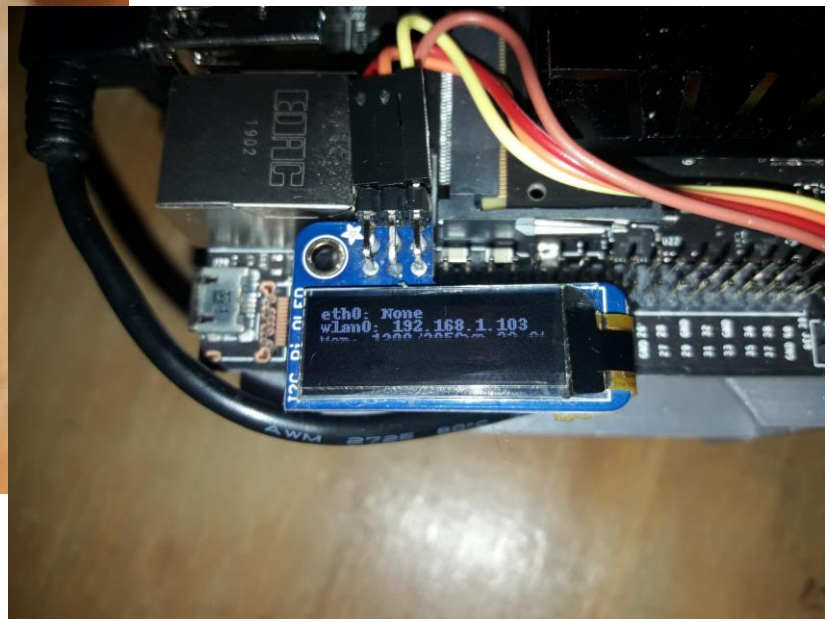




https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

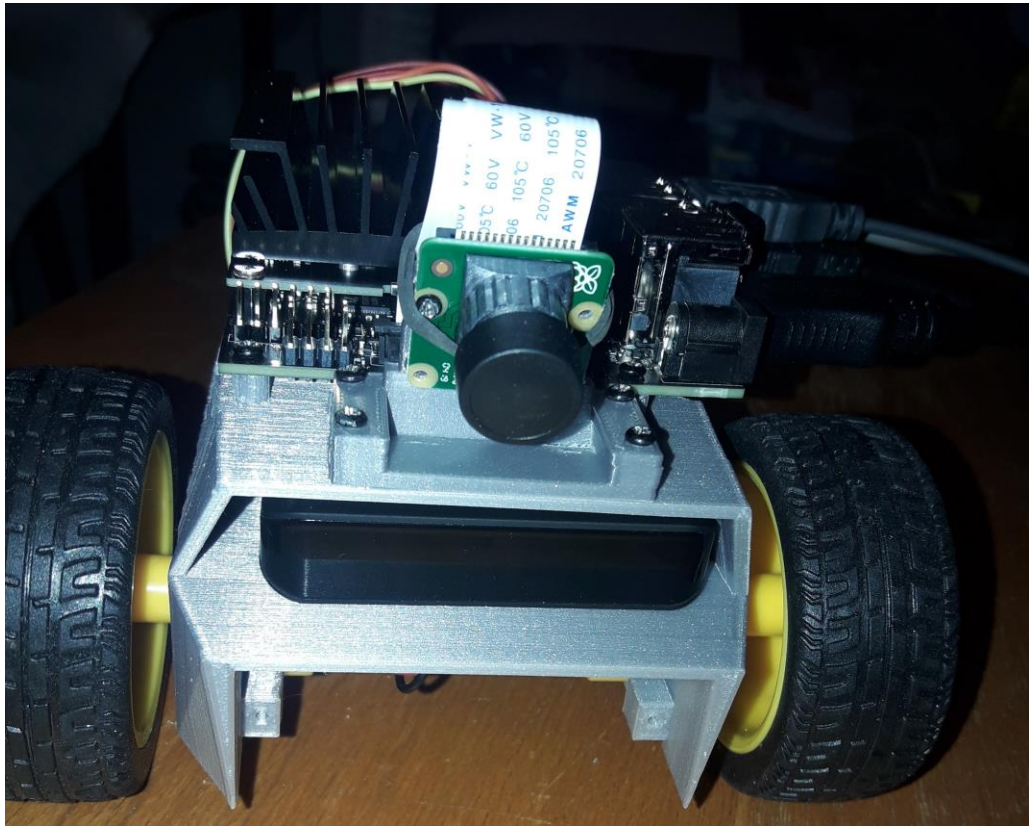
Montáž





https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Montáž



A robot je hotový. Ostáva zasunúť microSD kartu s OS a softwarom

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/hardware-setup

Pamäťová karta

SD Card 64GB – musí ich mať naozaj! Nie Kingstone (najlacnejšia možnosť), vhodná je napríklad SanDisk Ultra microSDXC UHS-1

Naformátujeme pomocou SDCardFormatter v5

Image prenesieme cez balenaEtcher

Najnovší image je jetbot_image_v0p4p0.zip

https://drive.google.com/open?id=1G5nw0o3Q6E08xZM99ZfzQAe7-qAXxzHN

https://github.com/NVIDIA-AI-IOT/jetbot/wiki/Create-SD-Card-Image-From-Scratch

Operačný systém

Ubuntu 18.04.3 LTS (Bionic Beaver)

na platforme arm64

Nevyžaduje prihlásenie
user jetbot
passwd pre sudo: jetbot
prístup cez ssh: jetbot / jetbot
použijeme LAN alebo prihlásime to do wifi

gedit

http://<ip-address>:8888

sudo nvpmodel –q sudo nvpmodel –m1 #5W
seahorse Login / Change password jetbot Continue 2x

CUDA

ls /usr/local/
bin cuda cuda-10.0 …
CUDA 10.0

ls /usr/lib/aarch64-linux-gnu
ls
CuDNN 7.6.3

python3
>>> import cv2
>>> cv2.__version__
4.1.1

Predinštalované nástroje

jetbot_image_v0p4p0

1.14.0

3.6.9

tf.keras 2.2.4

4.1.1

torch 1.3.0
torchvision 0.4.0

ale nie 4.2 !

Doinštalujeme

```
sudo apt install libjpeg-dev libpng-dev libtiff-dev
sudo apt install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
sudo apt install libxvidcore-dev libx264-dev
sudo apt install libgtk-3-dev
sudo apt-get install libcanberra-gtk-module:arm64
sudo pip install –upgrade pip
```

Motory

```
# motor
from Adafruit_MotorHAT import Adafruit_MotorHAT as HAT
import time

left_motor_channel = 1 # right_motor_channel = 2
driver = HAT(i2c_bus=1)
motor = driver.getMotor(left_motor_channel)
speed = 200 # 0 .. 255
motor.setSpeed(speed)

motor.run(HAT.FORWARD)
time.sleep(0.6)

motor.run(HAT.BACKWARD)
time.sleep(0.6)

motor.run(HAT.RELEASE)
```

Pozor! Spúšťame python3 nie python!

Kamera

```
# Camera
#
# ls -l /dev/video0 must work, then install
# sudo apt-get install libcanberra-gtk-module:arm64
# to avoid error message 'Failed to load module "canberra-gtk-module"'
# or ignore it, the error message makes no harm

import cv2
```

OpenCV v JetBot-e používa Gstreamer

Ten umožňuje definovať zložitejšiu pipeline, ktorá spracúva obraz z kamery efektívnejšie.

Spravidla chceme upraviť snímaný obraz 1280 x 720 alebo na vhodný vstup neurónovej siete, napr. 224 x 224 a hľadáme framerate, ktorý stíhame spracovať.

Kamera

```
def gstreamer_pipeline(
    capture_width=1280, capture_height=720,
    display_width=1280, display_height=720,
    framerate=60, flip_method=0,
):
    return (
        "nvarguscamerasrc ! "
        "video/x-raw(memory:NVMM), "
        "width=(int)%d, height=(int)%d, "
        "format=(string)NV12, framerate=(fraction)%d/1 ! "
        "nvvidconv flip-method=%d ! "
        "video/x-raw, width=(int)%d, height=(int)%d, format=(string)BGRx ! "
        "videoconvert ! "
        "video/x-raw, format=(string)BGR ! appsink"
        % (
            capture_width, capture_height,
            framerate, flip_method,
            display_width, display_height,
        )
    )
```

Kamera

```
gp = gstreamer_pipeline(1280,720,224,224,25,0)
camera = cv2.VideoCapture(gp, cv2.CAP_GSTREAMER)

while True:
    re, image = camera.read()
    if not re:
        break

    #print(image.shape[1],'x',image.shape[0])
    cv2.imshow('camera',image)
    key = cv2.waitKey(1)
    if key == 27:
        break

cv2.destroyAllWindows()
```

Display

```
# 128x32 display with hardware I2C:
import Adafruit_SSD1306
from PIL import Image, ImageDraw, ImageFont
from jetbot.utils.utils import get_ip_address
disp = Adafruit_SSD1306.SSD1306_128_32(rst=None, i2c_bus=1, gpio=1)
disp.begin()
disp.clear()
disp.display()

width = disp.width
height = disp.height
image = Image.new('1', (width, height))
draw = ImageDraw.Draw(image)
draw.rectangle((0,0,width,height), outline=0, fill=0)
font = ImageFont.load_default()
#draw.text((0,0),"eth0: "+str(get_ip_address('eth0')),font=font,fill=255)
draw.text((0,8),"wlan0: "+str(get_ip_address('wlan0')),font=font,fill=255)
draw.text((0,16),"hallo",font=font,fill=255)
disp.image(image)
disp.display()
```

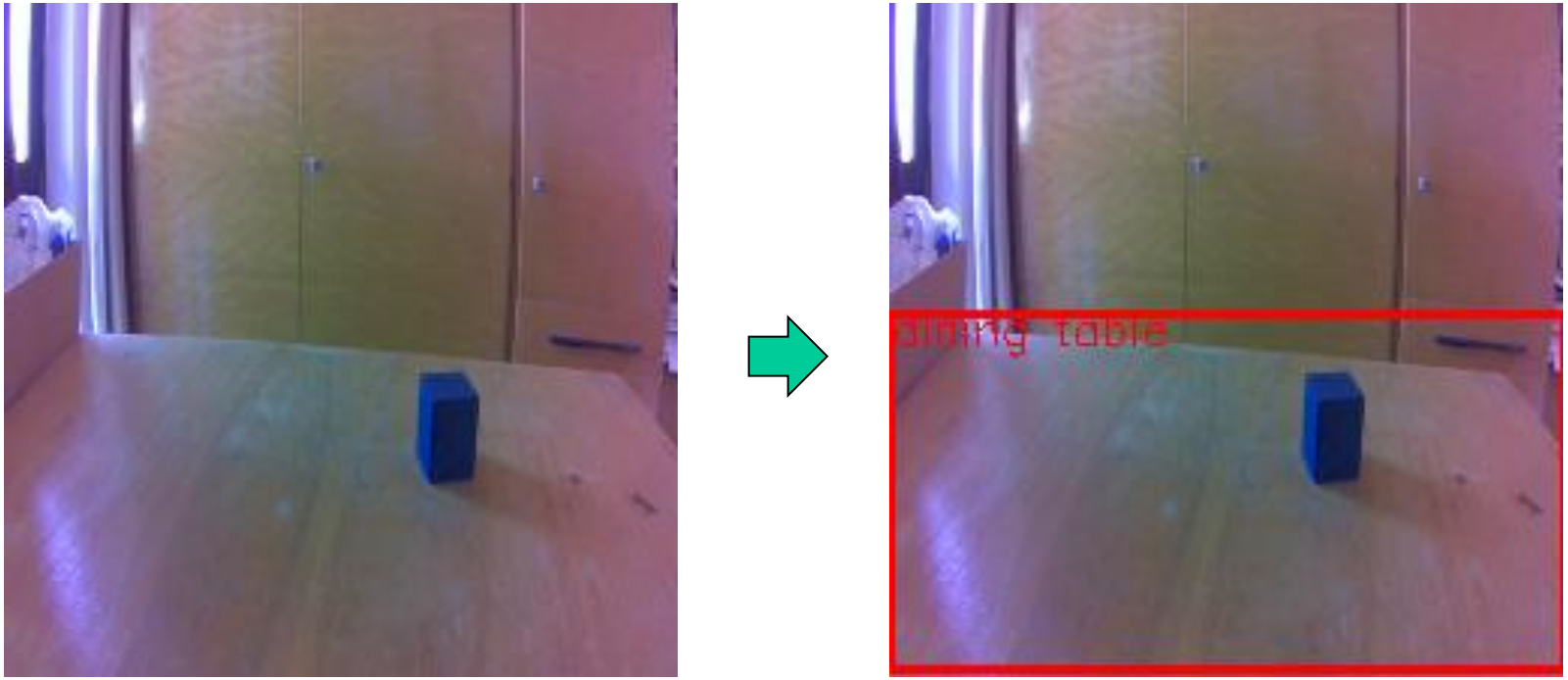
Edge AI

```
# use model trained by deep learnig:
import torchvision
from torchvision import transforms as T
from PIL import Image
import cv2

coco_labels = [
    '__background__', 'person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus',
    'train', 'truck', 'boat', 'traffic light', 'fire hydrant', 'N/A', 'stop sign',
    'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
    'elephant', 'bear', 'zebra', 'giraffe', 'N/A', 'backpack', 'umbrella', 'N/A', 'N/A',
    'handbag', 'tie', 'suitcase', 'frisbee', 'skis', 'snowboard', 'sports ball',
    'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard', 'tennis racket',
    'bottle', 'N/A', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl',
    'banana', 'apple', 'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza',
    'donut', 'cake', 'chair', 'couch', 'potted plant', 'bed', 'N/A', 'dining table',
    'N/A', 'N/A', 'toilet', 'N/A', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
    'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'N/A', 'book',
    'clock', 'vase', 'scissors', 'teddy bear', 'hair drier', 'toothbrush'
]

model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
model.eval()
model.cuda()
```

Edge AI

```
image = cv2.imread("../dataset/photo10.png")
img = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
img = Image.fromarray(img)
transform = T.Compose([T.ToTensor()])
img = transform(img)
img = img.cuda()

pred = model([img])

labels = [coco_labels[i] for i in list(pred[0]['labels'].cpu().numpy())]
boxes = [[(int(i[0]), int(i[1])), (int(i[2]), int(i[3]))] for i in
list(pred[0]['boxes'].cpu().detach().numpy())]
scores = list(pred[0]['scores'].cpu().detach().numpy())

threshold = 0.3
for label, box, score in zip(labels, boxes, scores):
    if score > threshold:
        cv2.rectangle(image,box[0],box[1],(0,0,255),2)
        cv2.putText(image,label,(box[0][0],box[0][1]+10),0,0.5,(0,0,255),1)

cv2.imshow('detection',image)
cv2.waitKey(0)
```

Edge AI

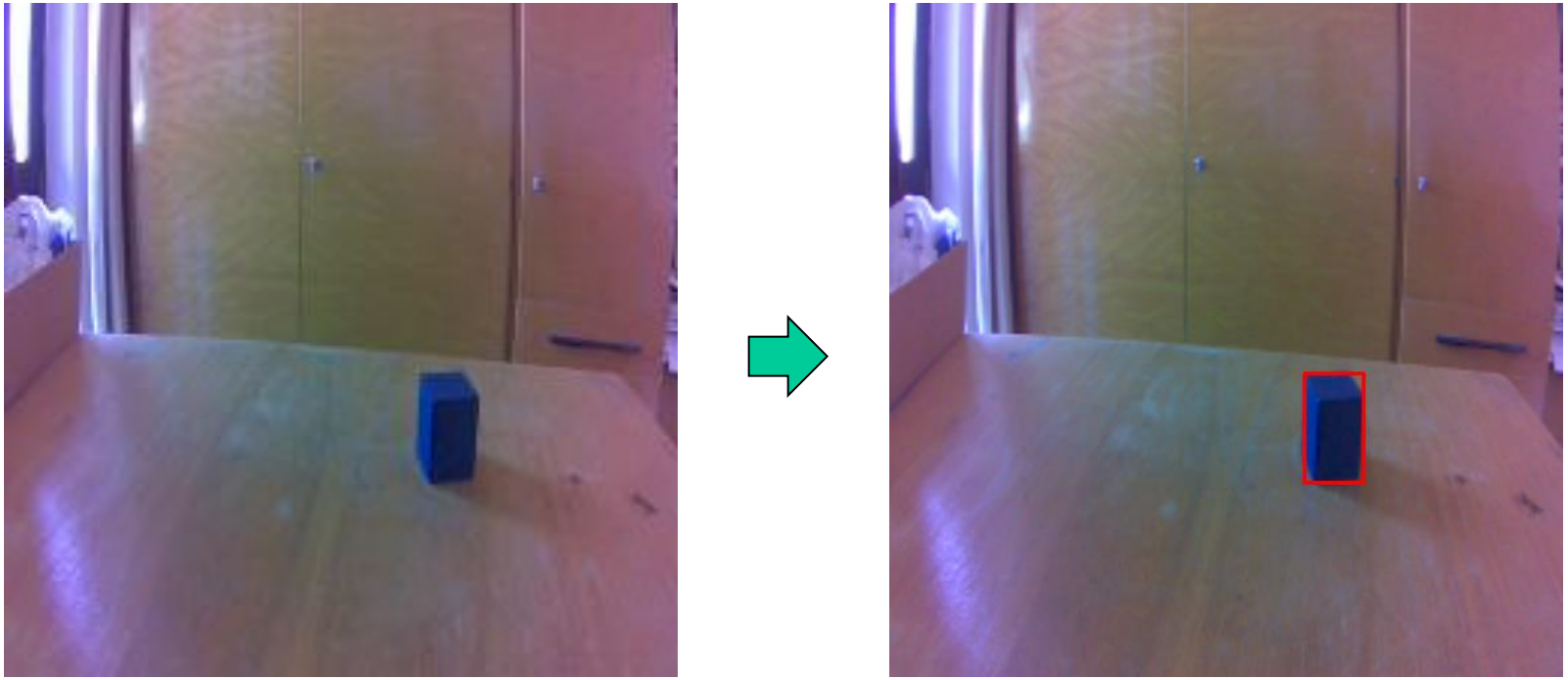


https://github.com/andylucny/JetBotDemos.git

Edge AI

```
import torch
from torchvision import transforms as T
import cv2
from PIL import Image
model_name = "trained_models/signatrix_efficientdet_coco.pth"
model = torch.load(model_name).module
model.cuda()
image = cv2.imread("../dataset/photo10.png")
image = cv2.resize(image,(512,512))
img = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
img = Image.fromarray(img)
transform = T.Compose([T.ToTensor()])
img = transform(img)
scores, labels, boxes = model(img.cuda().float().unsqueeze(dim=0))
if boxes.shape[0] > 0:
    for box_id in range(boxes.shape[0]):
        pred_prob = float(scores[box_id])
        if pred_prob > 0.3:
            pred_label = int(labels[box_id])
            xmin, ymin, xmax, ymax = boxes[box_id, :]
            cv2.rectangle(image, (xmin, ymin), (xmax, ymax), (0,0,255),2)
cv2.imwrite("prediction.png", image)
```

Edge AI



https://github.com/andylucny/JetBotDemos.git

Ďakujem za pozornosť!

Seminár Robotika.SK

JetBot

www.robotika.sk/seminar-archiv.php#prednaska37

Andrej Lúčny

Katedra aplikovanej informatiky FMFI UK

lucny@fmph.uniba.sk

http://dai.fmph.uniba.sk/w/Andrej_Lucny