



Introduction to Embedded Systems, Lecture 12

Alternative approaches to ES design:
RP Pico SDK, RTOS, FPGA, and AI powered ES

Dmitry Zaitsev

<http://daze.ho.ua>

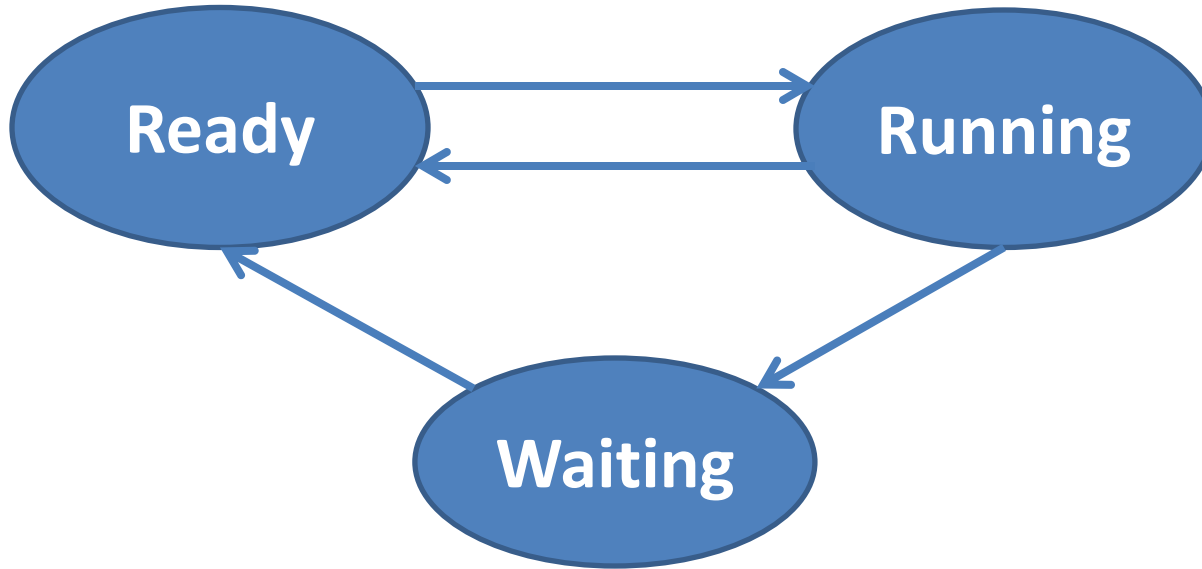
RP Pico SDK in Linux

- <https://github.com/raspberrypi/pico-sdk>
- Command line or IDE professional toolset for ES design
- C, C++ or assembly language
- Debugging facilities
- Rich set of libraries
- Possibility of linking with OS, for instance FreeRTOS, <https://www.freertos.org>

Operating system vs Bare hardware

- Operating system (RTOS) advantages:
 - OS provides dispatcher of processes, control of memory, devices, and information, inter-process communication
 - ES is implemented as a set of communicating concurrent processes
 - Waiting for an event blocks the current process and OS switches to other ready process

Typical process state diagram of OS



Message Exchange Example, 1

```
void vSenderTask(void *pvParameters) {  
    MessageBufferHandle_t buffer = (MessageBufferHandle_t) pvParameters;  
    char message[] = "FreeRTOS + Pi Pico";  
    for (;;) {  
        xMessageBufferSend(buffer, (void *)message, strlen(message), 0);  
        vTaskDelay(1000);  
    }  
}
```

Message Exchange Example, 2

```
void vReceiverTask(void *pvParameters) {  
    MessageBufferHandle_t buffer = (MessageBufferHandle_t) pvParameters;  
    size_t messageSize = BUFFER_SIZE - 4;  
    char *message = malloc(messageSize);  
    memset(message, '\0', messageSize);  
    size_t lengthReceived;  
    for (;;) {  
        lengthReceived = xMessageBufferReceive(buffer, (void *)message, BUFFER_SIZE, 0);  
        if (lengthReceived > 0) {  
            printf("length: %d, message: %s\n", lengthReceived, message);  
            memset(message, '\0', messageSize);  
        }  
    }  
}
```

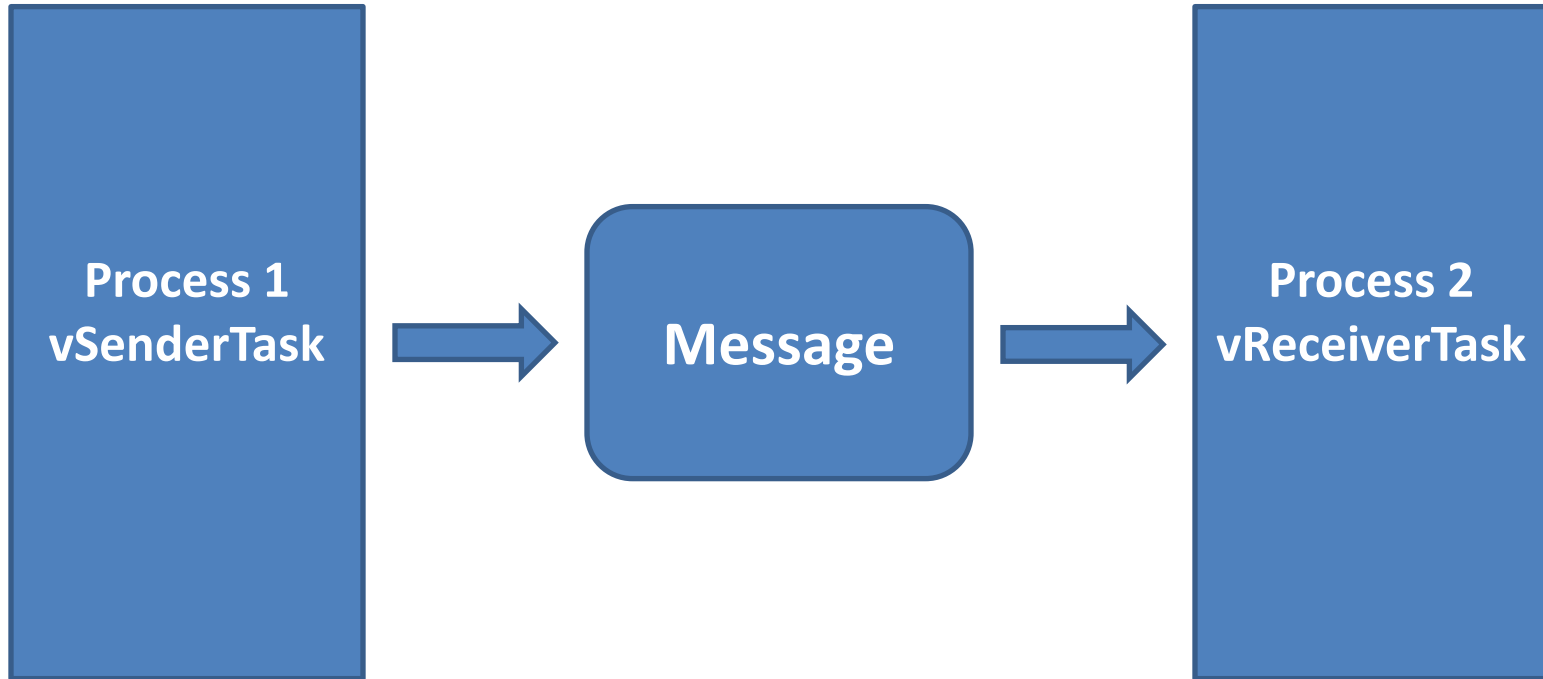
Message Exchange Example, 3

```
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include "pico/stdlib.h"
#include "FreeRTOS.h"
#include "task.h"
#include "message_buffer.h"

const size_t BUFFER_SIZE = 32;

void main() {
    stdio_init_all();
    busy_wait_ms(1000);
    MessageBufferHandle_t buffer = xMessageBufferCreate(BUFFER_SIZE);
    xTaskCreate(vSenderTask, "Sender", 128, (void *)buffer, 1, NULL);
    xTaskCreate(vReceiverTask, "Receiver", 128, (void *)buffer, 1, NULL);
    vTaskStartScheduler();
}
```

Message Exchange Example, 4



ES Design with FPGA

- Advantages of hard-wired control together with microcontroller's flexibility
- Completely concurrent concept
- Low latency control
- Using well supported digital hardware design technology based on HDL, say Verilog
- Just add mapping of pins on Verilog module input-output parameters

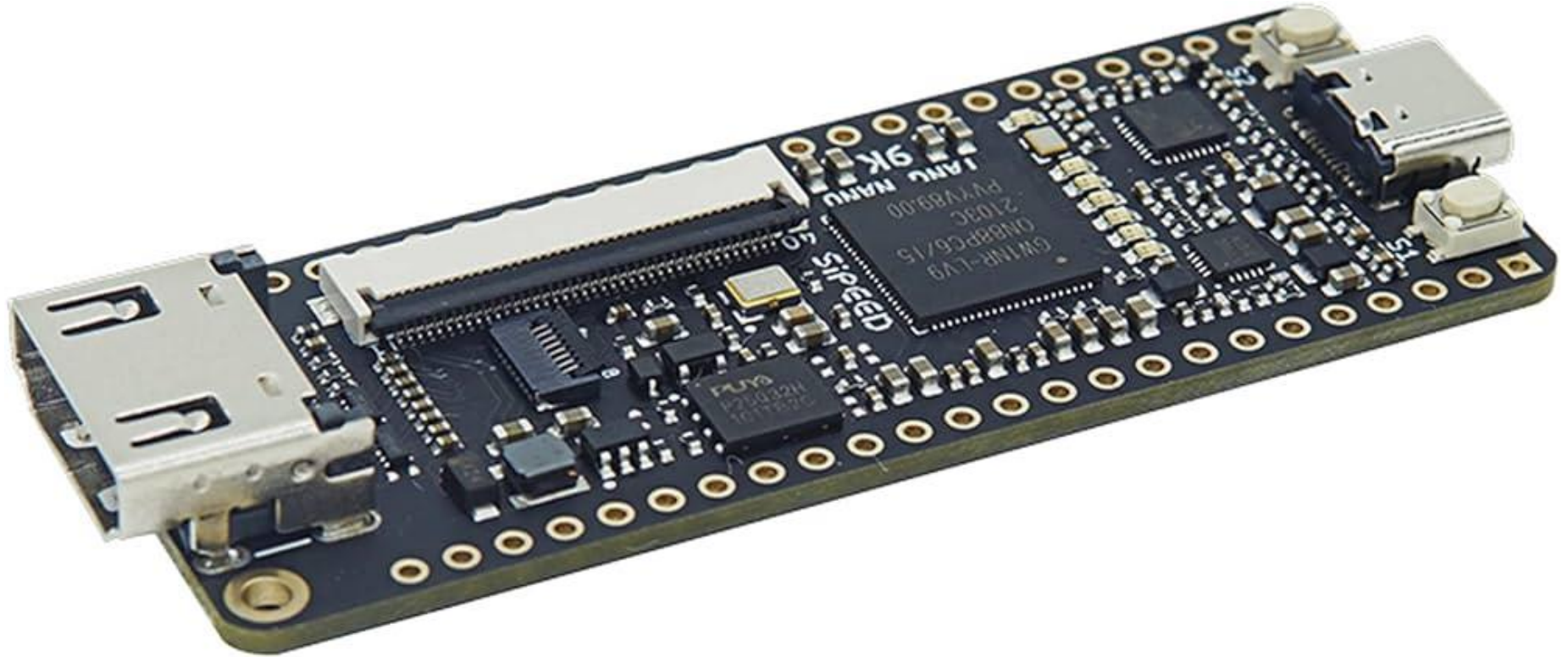
What is FPGA?

- Field Programmable Gate Arrays (FPGAs) are integrated circuits often sold off-the-shelf.
- They're referred to as 'field programmable' because they provide customers the ability to reconfigure the hardware to meet specific use case requirements after the manufacturing process.
- FPGAs contain configurable logic blocks (CLBs) and a set of programmable interconnects that allow the designer to connect blocks and configure them to perform everything from simple logic gates to complex functions.

FPGA Manufactures

- Advanced Micro Devices, Inc. (US), Xilinx
- GOWIN Semiconductor Corporation (China)
- Intel Corporation (US), Altera
- Microchip Technology Inc., (US)
- ByteSnap Design (UK)

Tang Nano 9K FPGA

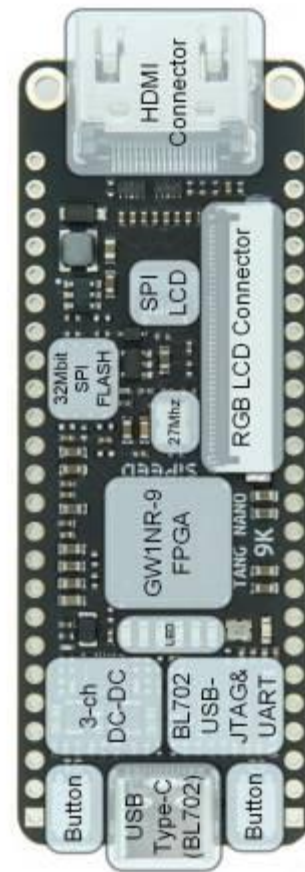


Tang Nano 9K FPGA Features, 1

- powered by Gowin's GW1NR-9 FPGA chip
- HDMI, RGB screen, and SPI screen interfaces, a 32Mbit SPI flash and six LEDs
- 8640 LUT4 logic units
- onboard 27MHz clock and 2 PLLs
- onboard USB-JTAG & USB-UART

Tang Nano 9K FPGA Features, 2

Logic units(LUT4)	8640
Registers(FF)	6480
ShadowSRAM SSRAM(bits)	17280
Block SRAM BSRAM(bits)	468K
Number of B-SRAM	26
User flash(bits)	608K
SDR SDRAM(bits)	64M
18 x 18 Multiplier	20
SPI FLASH	32M-bit
Number of PLL	2





Process

- Design Summary
- ✓ User Constraints
 - FloorPlanner
 - Timing Constraints Editor
- ✓ Synthesize
 - Synthesis Report
 - Netlist File
- ✓ Place & Route
 - Place & Route Report
 - Timing Analysis Report
 - Ports & Pins Report
 - Power Analysis Report
- Programmer

General

Project File:	C:\Gowin\Gowin_V1.9.9_x64\IDE\bin\Documents\led-btn\led-btn.gprj
Synthesis Tool:	GowinSynthesis

Target Device

Part Number:	GW1NR-LV9QN88PC6/I5
Series:	GW1NR
Device:	GW1NR-9
Device Version:	C
Package:	QFN88P
Speed Grade:	C6/I5
Core Voltage:	LV

Start Page

Console



About



—智慧逻辑 定制未来—

GOWIN FPGA Designer
Version V1.9.9 (64-bit) build(69780)

Copyright © 2014-2024 GOWINSEMI
All Rights Reserved

OK

Button – LED Project

GOWIN FPGA Designer - [C:\Gowin\Gowin_V1.9.9_x64\IDE\bin\Documents\led-btn\src\led-btn-verilog.v *]

File Edit Project Tools Window Help



Design

led-btn - [C:\Gowin\Gowin_V1.9.9_x64\IDE\bin\Documents\led-btn\led-btn.gprj]

GW1NR-LV9QN88PC6/I5

Verilog Files

src\led-btn-verilog.v

Physical Constraints Files

src\led-btb-constr.cst

```
1 module led_btn (  
2     input btn_i,  
3     output led_o  
4 );  
5  
6  
7     assign led_o = btn_i;  
8  
9 endmodule  
10
```

Design

led-btn - [C:\Gowin\Gowin_V1.9.9_x64\IDE\bin\Documents\led-btn\led-btn.gprj]

GW1NR-LV9QN88PC6/I5

Verilog Files

src\led-btn-verilog.v

Physical Constraints Files

src\led-btb-constr.cst

```
1 IO_LOC "btn_i" 3;  
2 IO_LOC "led_o" 11;  
3
```


Design Summary

- User Constraints:
 - Floor Planner
 - Timing Constraints Editor
- Synthesize (on Verilog)
- Place & Route
- Programmer

Programming FPGA

GOWIN FPGA Designer - [C:\Gowin\Gowin_V1.9.9_x64\IDE\bin\Documents\led-btn\src\led-btn-verilog.v *]

File Edit Project Tools Window Help

Process

- Design Summary
- User Constraints
 - FloorPlanner
 - Timing Constraints Editor
- Synthesize
 - Synthesis Report
 - Netlist File
- Place & Route
 - Place & Route Report
 - Timing Analysis Report
 - Ports & Pins Report
 - Power Analysis Report
- Programmer

```
1 module led_btn (  
2     input btn_i,  
3     output led_o  
4 );  
5  
6  
7     assign led_o = btn_i;  
8  
9 endmodule  
10
```

Gowin Programmer Version 1.9.9 (64-bit) build 31221

File Edit Tools About

USB Cable Setting

Enable	Series	Device	Operation	FS File	User Code
1 <input checked="" type="checkbox"/>	GW1NR	GW1NR-9C	SRAM Program	C:/Gowin/Gowin_V1.9.9_x64/IDE/bin/Documents/led-...	0x00000669

Progress information

Programming...

55%

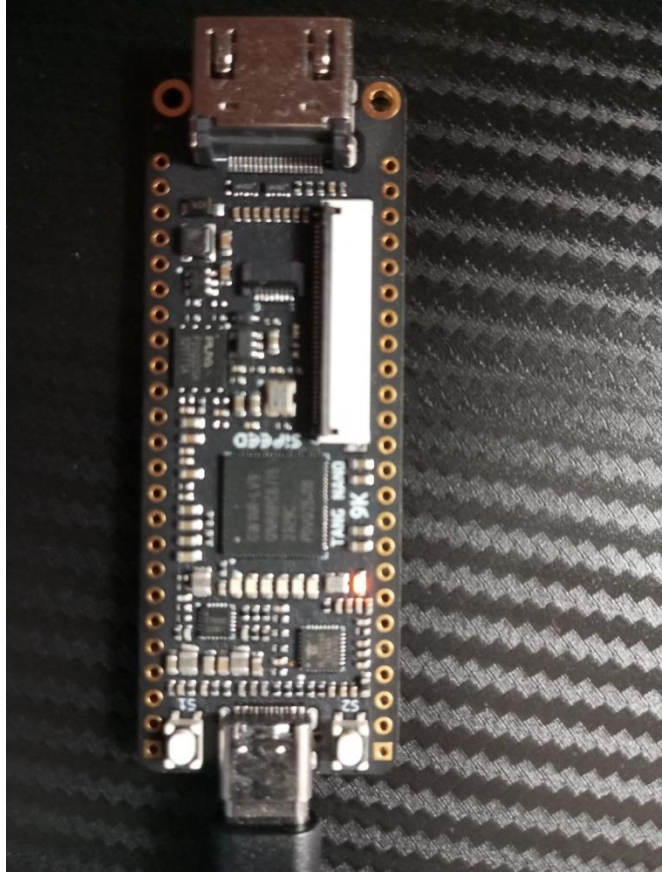
Cancel

Output

Info Target Device: GW1NR-9C(0x1100481b)
Info Operation "SRAM Program" for device#1...
Info User Code: 0x00000669
Info Status Code: 0x0003F020
Info Cost 3.92 second(s)
Info Target Cable: USB Debugger A/0/273/null@2.5MHz

Windows taskbar: Type here to search, 4:36 PM, 4/22/2024

How it works

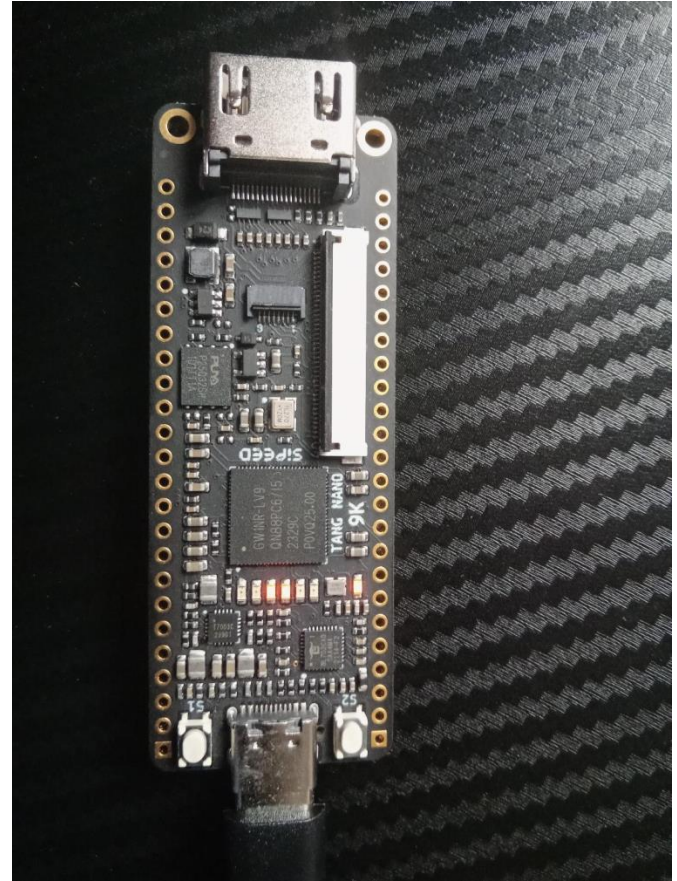
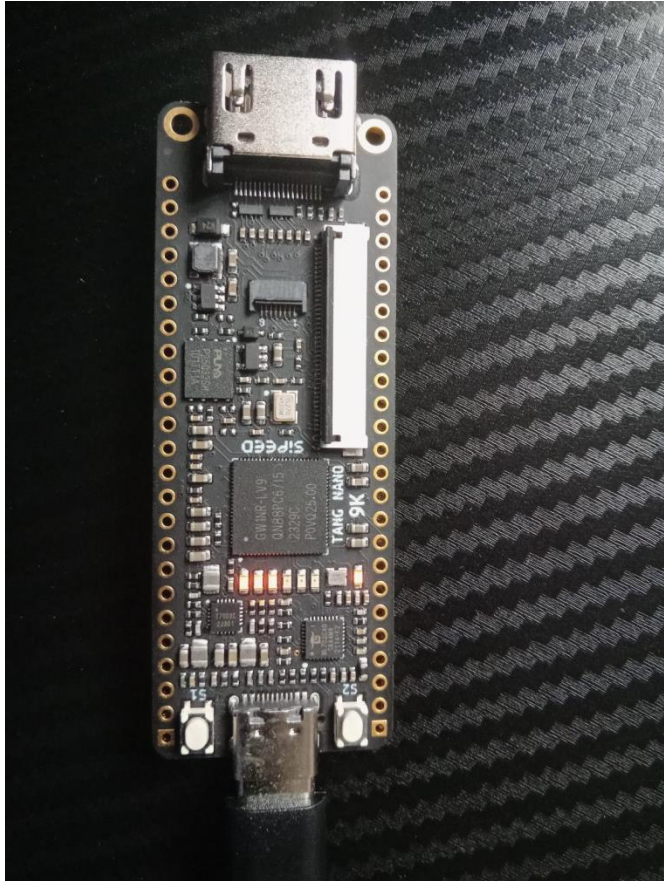


An example of LED counter

```
module top (  
    input clk,  
    output [5:0] led );  
localparam WAIT_TIME = 13500000;  
reg [5:0] ledCounter = 0;  
reg [23:0] clockCounter = 0;  
always @(posedge clk) begin  
    clockCounter <= clockCounter + 1;  
    if (clockCounter == WAIT_TIME) begin  
        clockCounter <= 0;  
        ledCounter <= ledCounter + 1;  
    end  
end  
assign led = ~ledCounter;  
endmodule
```

```
IO_LOC "clk" 52;  
IO_PORT "clk" PULL_MODE=UP;  
IO_LOC "led[0]" 10;  
IO_LOC "led[1]" 11;  
IO_LOC "led[2]" 13;  
IO_LOC "led[3]" 14;  
IO_LOC "led[4]" 15;  
IO_LOC "led[5]" 16;
```

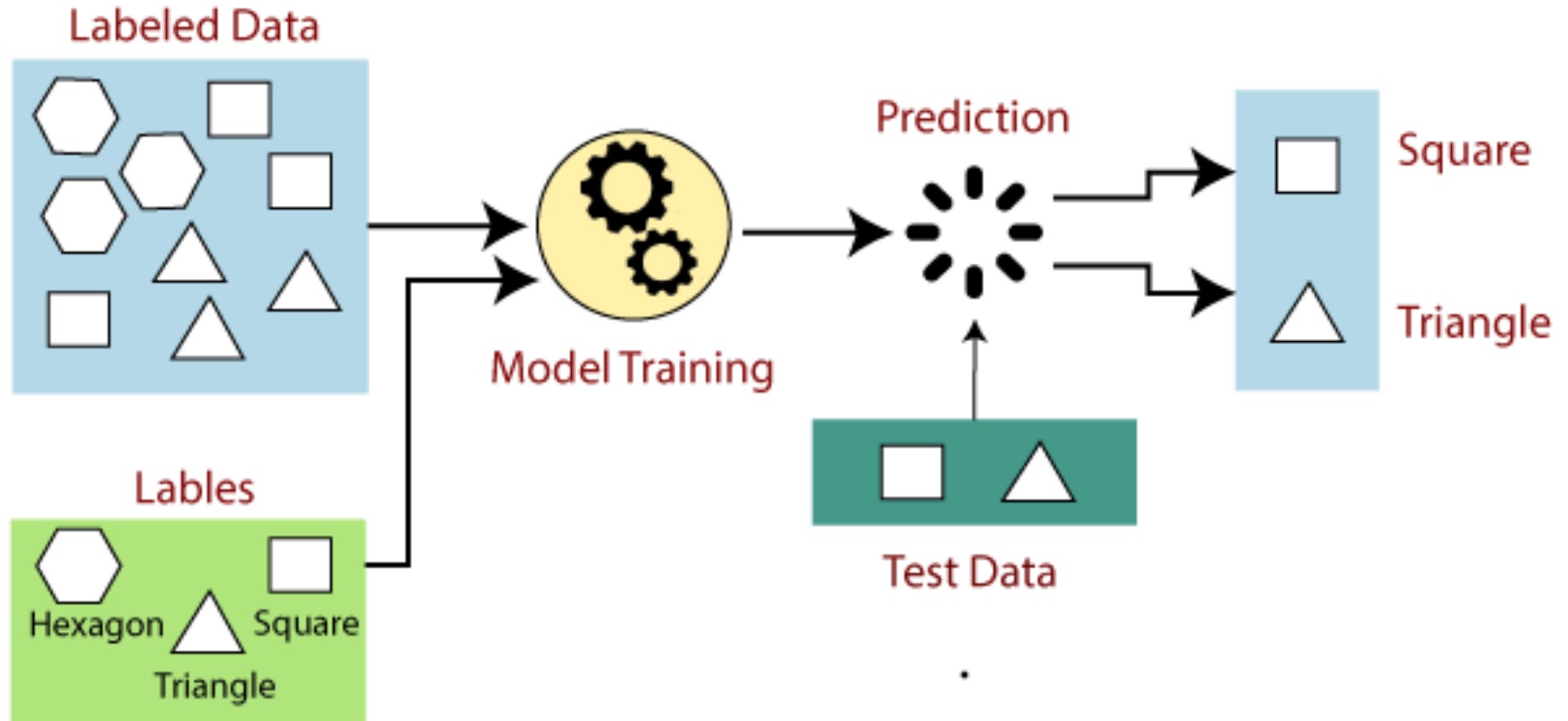
LED counter view



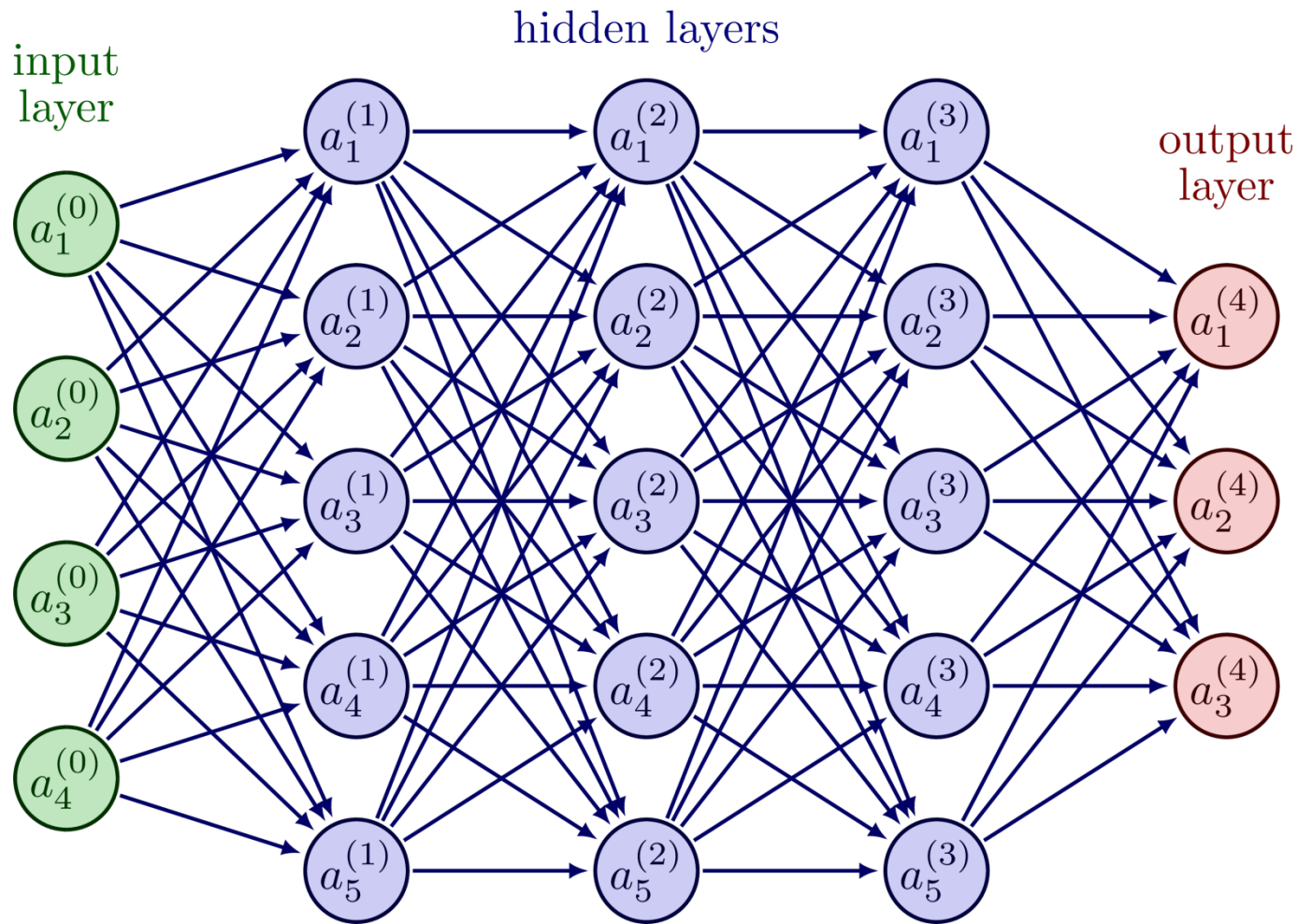
Artificial Intelligence Powered ES

- Train neural network (NN) on a data series of examples for basic tasks of ES
- Use NN instead of ES software
- Run NN on a dedicated hardware
- Applications: self-driving vehicles, face recognition, car number plates recognition, etc

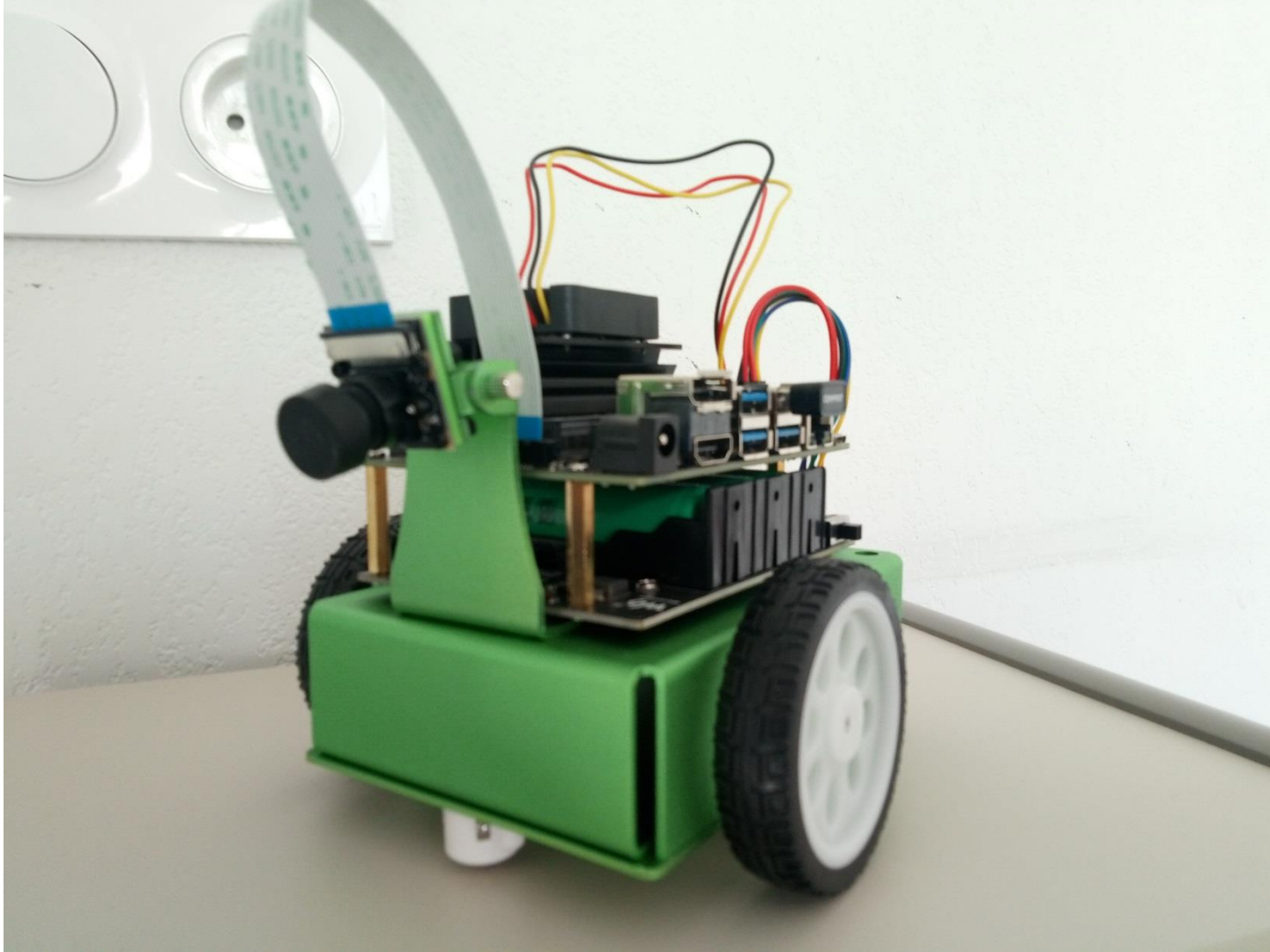
Supervised Machine Learning Example



Machine learning neural network architecture



Example: JetBot powered by NVIDIA Jetson Nano



Parts to assemble



NVIDIA Jetson Nano – computer for embedded AI applications



Jetson Nano Technical Specifications

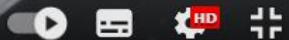
GPU	128-core Maxwell
CPU	Quad-core ARM A57 @ 1.43 GHz
Memory	4 GB 64-bit LPDDR4 25.6 GB/s
Storage	16 GB eMMC 5.1
Connectivity	Gigabit Ethernet
Display	HDMI 2.0, eDP 1.4, DP 1.2
PCIe	1x1/2/4 PCIe Gen2
USB	1x USB 3.0, 3x USB 2.0
Others	I ² C, I ² S, SPI, UART, SD/SDIO, GPIO

Jetbot Road Following and Collision Avoidance



0:22 / 0:57

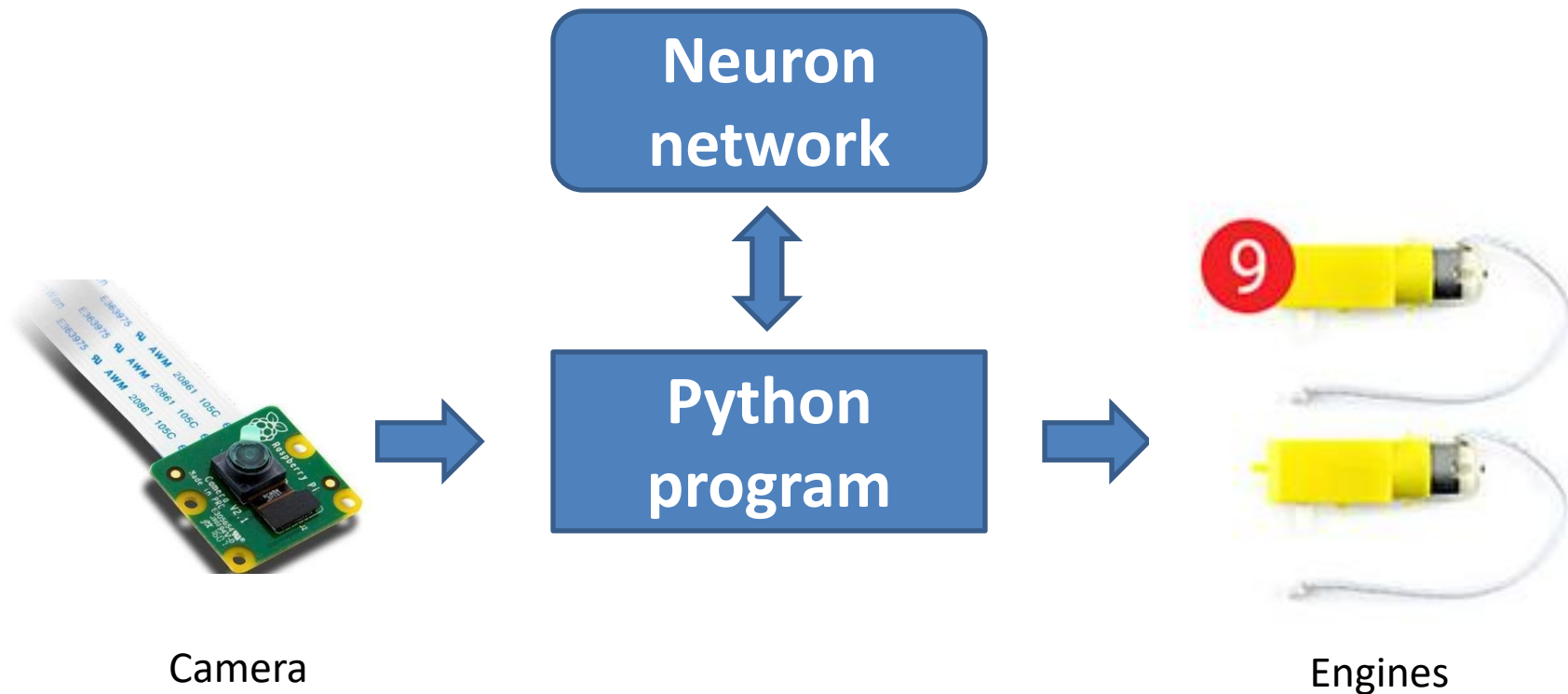
Прокрутите экран вниз, чтобы посмотреть подробную информацию



How it works

- Autonomous vehicle follows road and avoids collisions
- Python program runs neuron network
- Python program provides connection of neuron network to camera and engines
- <https://youtu.be/8Hz2G2SK3KI>

Python and AI based control



How to do

- Test NANO Jetson Toolkit
- Assemble JetBot
- Collect images/video for collision avoidance or/and road following
- Train neuron network
- Run neuron network to drive JetBot

Python

Jupyter notebooks for JetBot

tesla Stock Predictions | Kaggle | Tesla, Inc. (TSLA) Stock Historical | jetbot/notebooks/collision_avoidance

github.com/NVIDIA-AI-IOT/jetbot/tree/master/notebooks/collision_avoidance

Product Solutions Open Source Pricing Search Sign in Sign up

NVIDIA-AI-IOT / jetbot Public Notifications Fork 933 Star 2.6k

<> Code Issues 13 Pull requests 6 Discussions Actions Projects Wiki Security Insights

master jetbot / notebooks / collision_avoidance / Go to file

tokk-nv Add missing descriptions be2ca63 on Jan 9, 2021 History

..		
data_collection.ipynb	Merge branch 'master' into fix-typos	2 years ago
live_demo.ipynb	Merge branch 'master' into fix-typos	2 years ago
live_demo_resnet18.ipynb	Add missing descriptions	2 years ago
live_demo_resnet18_build_trt.ipynb	Split Collision Avoidance TRT inference notebook	2 years ago
live_demo_resnet18_trt.ipynb	Add missing descriptions	2 years ago
train_model.ipynb	Merge branch 'dev-headroom' of https://github.com/tokk-nv/jetbot into...	2 years ago
train_model_plot.ipynb	Fix typos in notebooks/collision_avoidance/	2 years ago
train_model_resnet18.ipynb	Change parameters to stabilize training across platforms	2 years ago

Activate Windows
Go to Settings to activate Windows.

Type here to search 55°F Cloudy 6:46 PM 1/7/2023

Collision Avoidance - Train Model

Welcome to this host side Jupyter Notebook! This should look familiar if you ran through the notebooks that run on the robot. In this notebook we'll train our image classifier to detect two classes `free` and `blocked`, which we'll use for avoiding collisions. For this, we'll use a popular deep learning library *PyTorch*

```
In [ ]: import torch
import torch.optim as optim
import torch.nn.functional as F
import torchvision
import torchvision.datasets as datasets
import torchvision.models as models
import torchvision.transforms as transforms
```

Upload and extract dataset

Before you start, you should upload the `dataset.zip` file that you created in the `data_collection.ipynb` notebook on the robot.

You should then extract this dataset by calling the command below

```
In [ ]: !unzip -q dataset.zip
```

You should see a folder named `dataset` appear in the file browser.

Create dataset instance

Now we use the `ImageFolder` dataset class available with the `torchvision.datasets` package. We attach transforms from the `torchvision.transforms` package to prepare the data for training.

```
In [ ]: dataset = datasets.ImageFolder(
```

Activate Windows
 Go to Settings to activate Windows.

Life critical decisions within ES

- AI based suggestion
- Algorithm based check
- Human being approval