SMALL IS GOING BIG: GO ON MICROCONTROLLERS

GOPHERCON 2019

RON EVANS - @DEADPROGRAM

RON EVANS (@DEADPROGRAM)

TECHNOLOGIST FOR HIRE

SALVADOR EVANS
(SON OF @DEADPROGRAM)

APPRENTICE

HYBRIDGROUP.COM

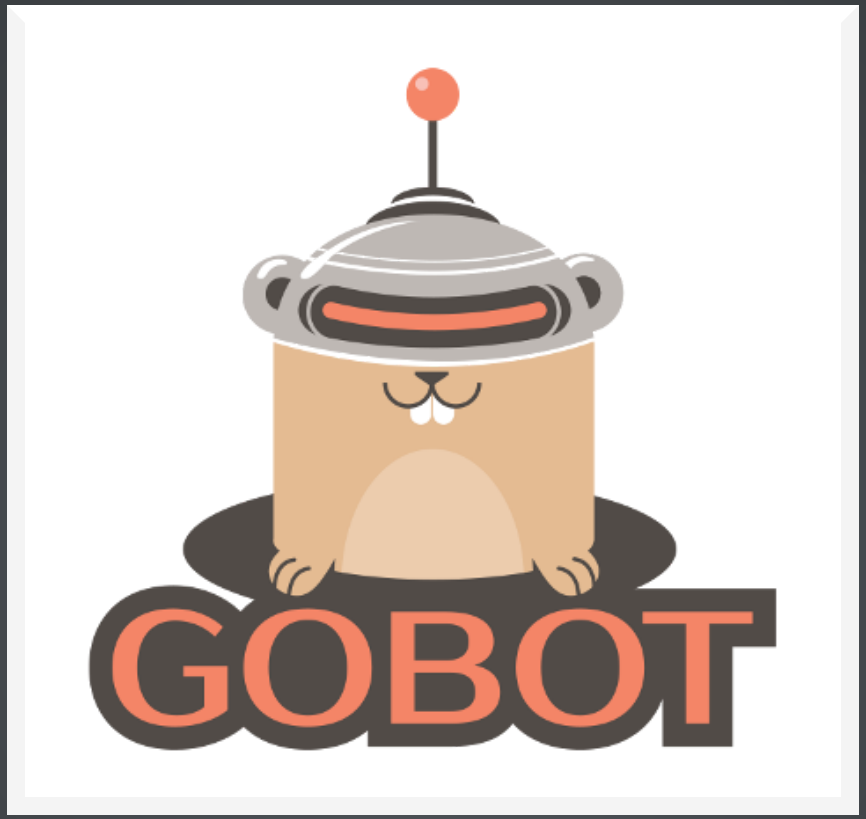
TECHNOLOGISTS FOR HIRE

CLIENTS

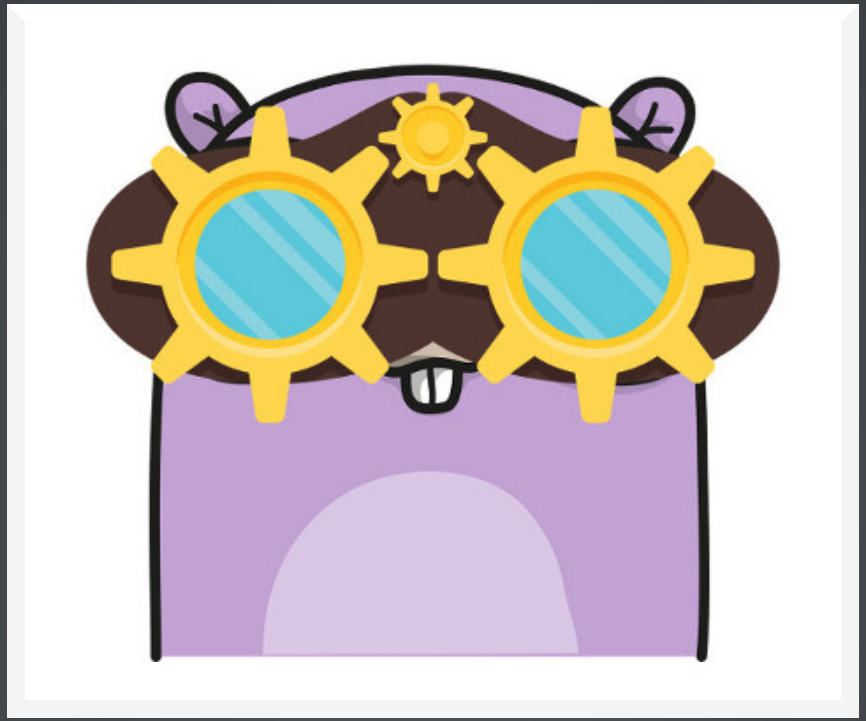

intel


STAR WARS
sphero

OPEN SOURCE PROJECTS



GOBOT.IO



GOCV.IO

TINYGO.ORG

GO FOR SMALL PLACES

EVERYONE SAYS GO IS TOO BIG

PROGRAMMING LANGUAGES OFTEN USED FOR EMBEDDED SYSTEMS

C

C++

PYTHON

JAVASCRIPT

RUST

GOLANG.ORG

BUT YOU SAID THAT EVERYONE SAYS THAT GO IS TOO BIG

I AM GOING TO SHOW YOU RIGHT NOW THAT GO CAN BE...

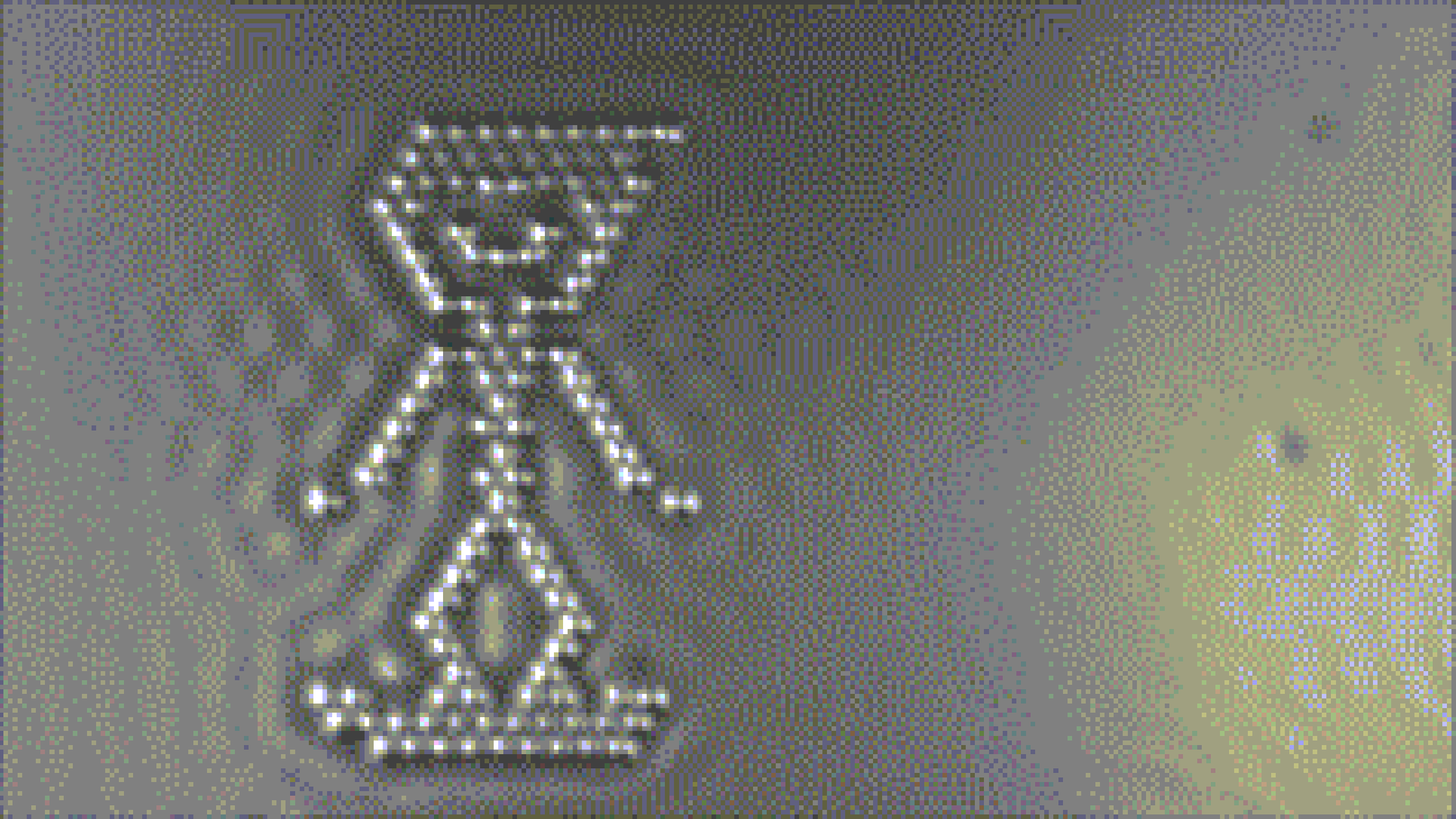
TINY

'HELLO, WORLD' IN GO

```
package main

func main() {
        println("Hello, world")
}
```

'HELLO, WORLD' USING GO 1.12
VS.
'HELLO, WORLD' USING TINYGO 0.7.0

HOW CAN YOU DO THAT?



LET'S JUST CLEAR UP ONE THING RIGHT NOW

TINYGO IS NOT THE SAME THING AS THE FULL GO

TINYGO DOES NOT SUPPORT THE ENTIRE GO LANGUAGE...

YET

TINYGO DOES NOT SUPPORT THE ENTIRE GO STANDARD LIBRARY...

YET

BUT TINYGO CAN DO A LOT ALREADY

AND TINYGO IS ALREADY VERY USEFUL FOR SMALL PLACES

MICROCONTROLLERS

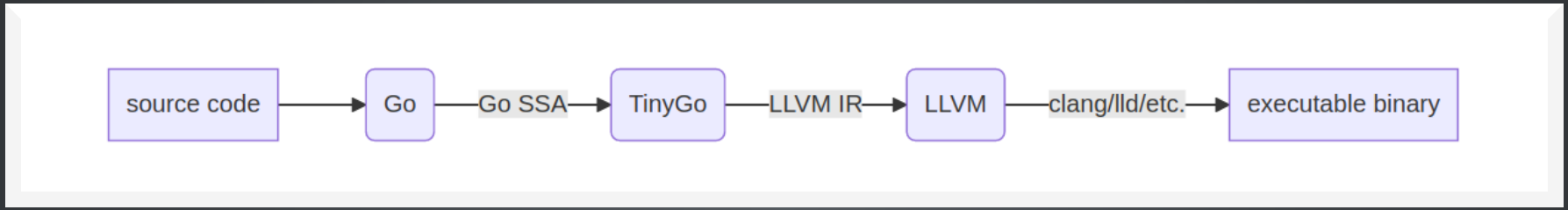
WEBASSEMBLY

HOW TINYGO WORKS

GO → TINYGO → LLVM

GO COMPILER TOOLCHAIN IS WRITTEN IN GO

LLVM - A SET OF TOOLS FOR BUILDING COMPILERS



TinyGo compiler architecture

'HELLO, WORLD' OF THINGS

DIGISPARK + LED

DIGISPARK

ATMEL ATTINY85

8-BIT PROCESSOR

16 MHZ

8K FLASH

```
package main

import (
        "machine"
        "time"
)

func main() {
        led := machine.LED
        led.Configure(machine.PinConfig{Mode: machine.PinOutput})
        for {
                led.Low()
                time.Sleep(time.Millisecond * 500)

                led.High()
                time.Sleep(time.Millisecond * 500)
        }
}
```

DEMO

GPIO INPUT & OUTPUT

ADAFRUIT CIRCUIT PLAYGROUND EXPRESS + LED + BUTTON

ADAFRUIT CIRCUIT PLAYGROUND EXPRESS

MICROCHIP ATSAMD21G18

ARM CORTEX M0

32-BIT PROCESSOR

48 MHZ

256K FLASH

```
package main

import (
        "machine"
        "time"
)

func main() {
        led := machine.A2
        led.Configure(machine.PinConfig{Mode: machine.PinOutput})

        button := machine.BUTTONA
        button.Configure(machine.PinConfig{Mode: machine.PinInput})

        for {
                if button.Get() {
                        led.High()
                } else {
                        led.Low()
                }

                time.Sleep(time.Millisecond * 10)
        }
}
```

DEMO

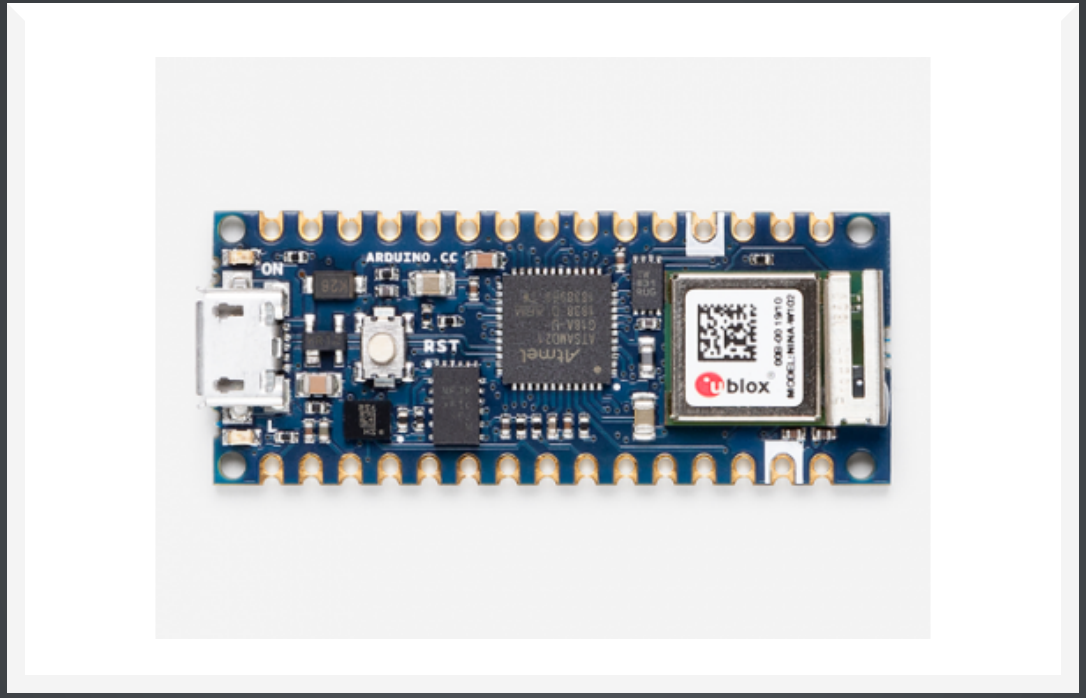
TOMORROW - COMMUNITY DAY

HARDWARE HACK SESSION

CHOOSE YOUR OWN HARDWARE ADVENTURE

TAKE HOME SOME COOL HARDWARE TO START PLAYING WITH TINYGO

TINYGO - POWERED BY ARDUINO



ARDUINO NANO33 IOT

DIVERSITY SCHOLARSHIP RECIPIENTS

PLEASE TWEET USING
#TINYGO #ARDUINO #GOPHERCON
TO THANK THEM!

INTERNET OF THINGS

INTERNET PROTOCOL (IP)

GO STANDARD LIBRARY

```
package net
```

NOT YET

WE DO HAVE A NUMBER OF COMPATIBLE INTERFACES

PAHO MQTT CLIENT

MQTT SENSOR STATION

ARDUINO NANO33 IOT + LED + BUTTON

ARDUINO NANO33 IOT

MICROCHIP ATSAMD21G18

ARM CORTEX M0

32-BIT PROCESSOR

48 MHZ

256K FLASH

UNIVERSAL ASYNCRONOUS RECEIVER/TRANSMITER (UART)

UBLOX NINA-W102

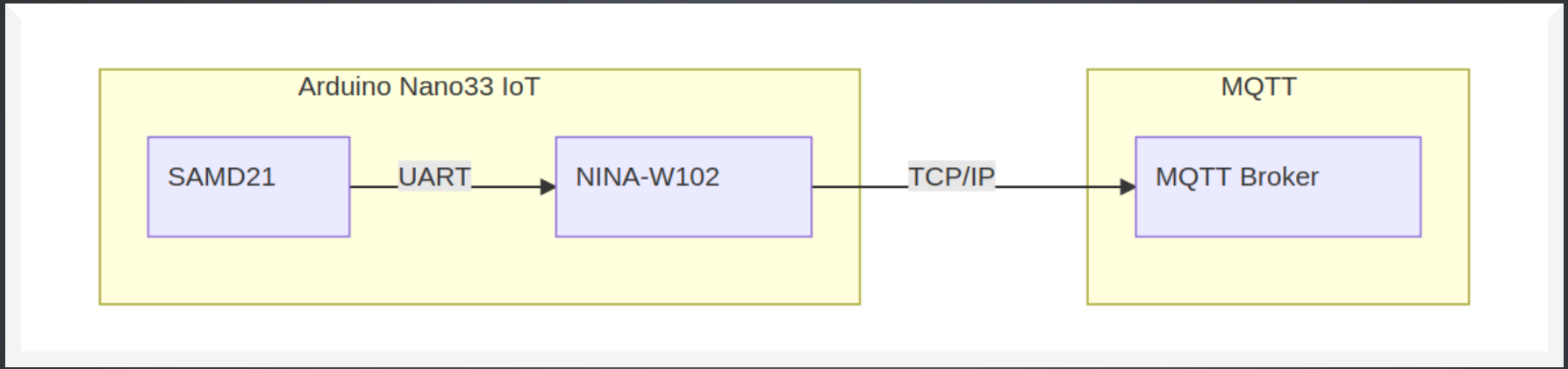
DUAL-CORE XTENSA LX6

32-BIT PROCESSOR

WIFI/BLUETOOTH LE

240 MHZ

16MB FLASH

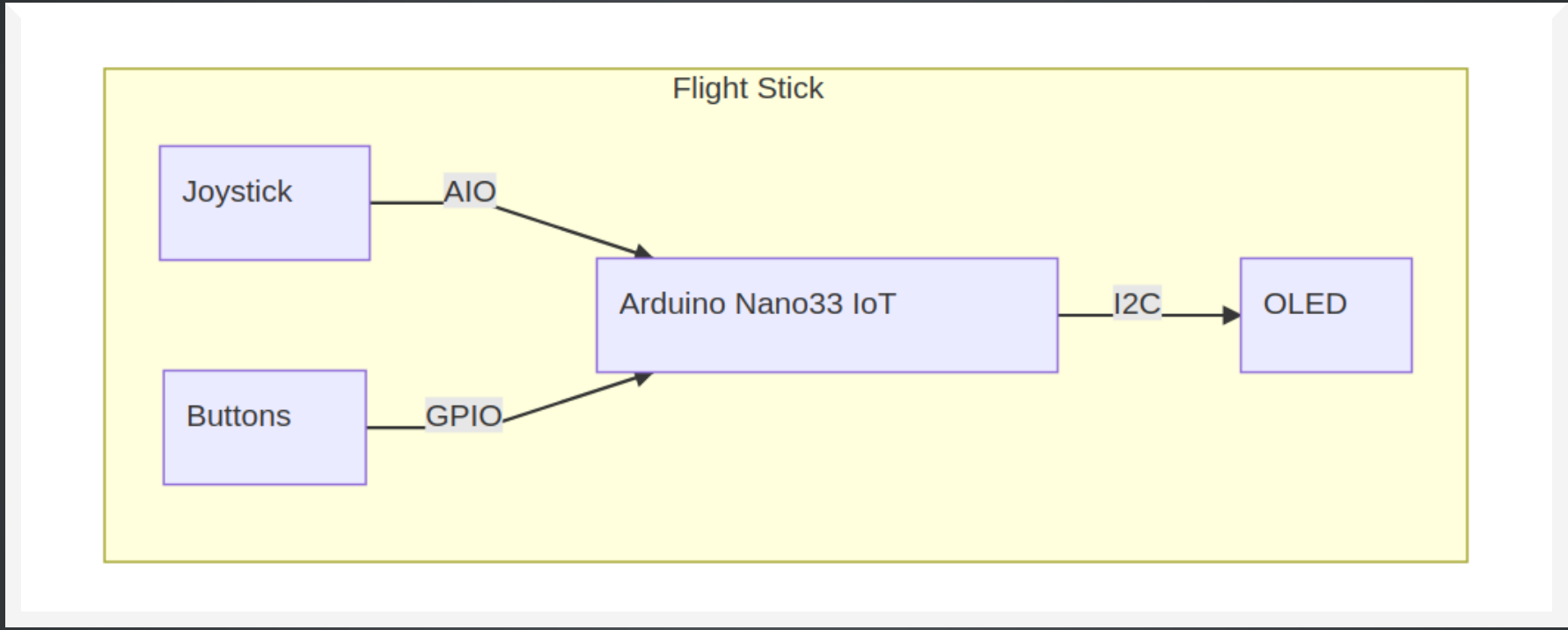

Arduino Nano33 IoT
SAMD21
UART
NINA-W102
TCP/IP
MQTT
MQTT Broker

```
package main

import (
        "machine"
        "math/rand"
        "time"

        "tinygo.org/x/drivers/espat"
        "tinygo.org/x/drivers/espat/mqtt"
)

// access point info
const ssid = "YOURSSID"
const pass = "YOURPASS"

// IP address of the MQTT broker to use. Replace with your own info.
// const server = "tcp://test.mosquitto.org:1883"
// const server = "ssl://test.mosquitto.org:8883"
const server = "tcp://10.42.0.1:1883"

// change these to connect to a different UART or pins for the ESP8266/ESP32
var (
        uart = machine.UART1
        tx   = machine.PA22
        rx   = machine.PA23

        adaptor *espat.Device
        topic   = "tinygo"
)
```

DEMO

FLIGHT CONTROLLER

ARDUINO NANO33 IOT + ANALOG JOYSTICK + OLED DISPLAY + BUTTONS + UART



Flight Control System architecture

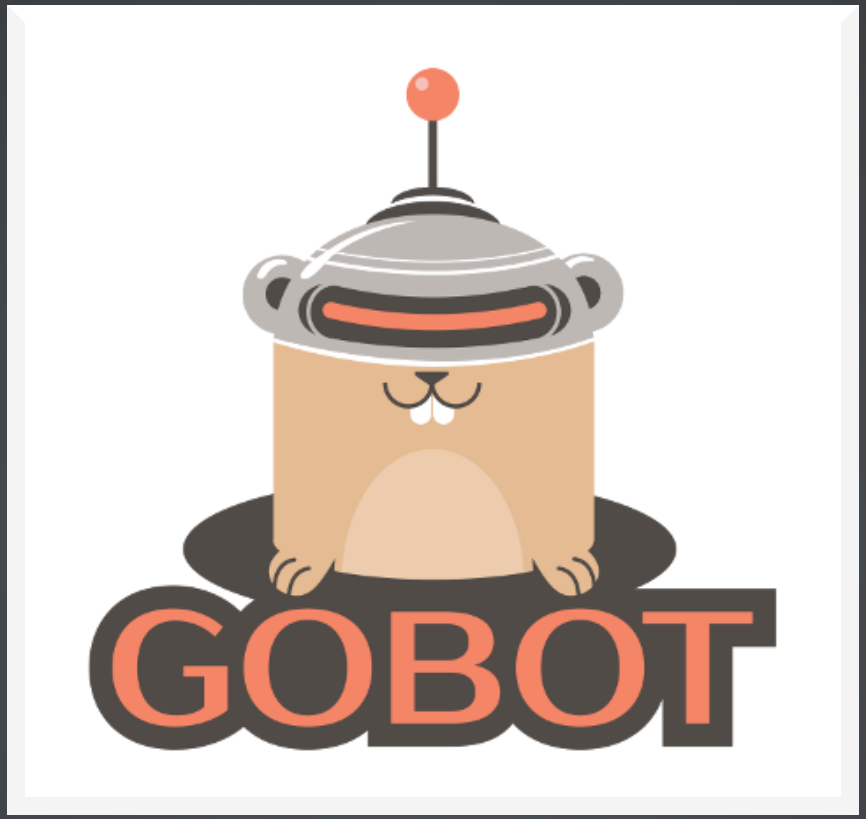
```
package main

import (
        "image/color"
        "machine"
        "strconv"
        "time"

        "github.com/conejoninja/tinydraw"
        "github.com/conejoninja/tinyfont"

        // comes from "github.com/conejoninja/tinyfont/freemono"
        freemono "./fonts"
        "tinygo.org/x/drivers/ssd1306"
)

var (
        xPos    uint16
        yPos    uint16
        b1push  bool
        b2push  bool
        b3push  bool
        b4push  bool
        joypush bool
)

func main() {
        machine.I2C0.Configure(machine.I2CConfig{
                Frequency: machine.TWI_FREQ_400KHZ,
```

DEMO

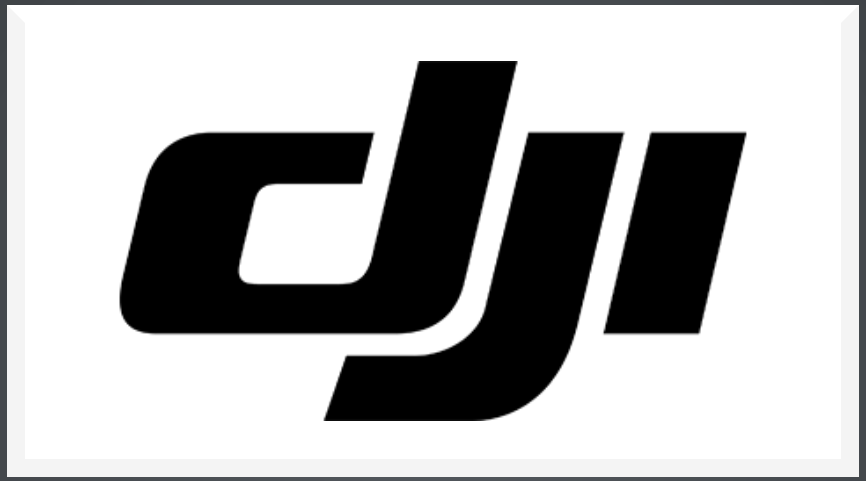
FULL APPLICATION: FLIGHT CONTROL

FLIGHT SYSTEM

FLIGHT CONTROLLER + CV GROUND SYSTEM + TELLO DRONE

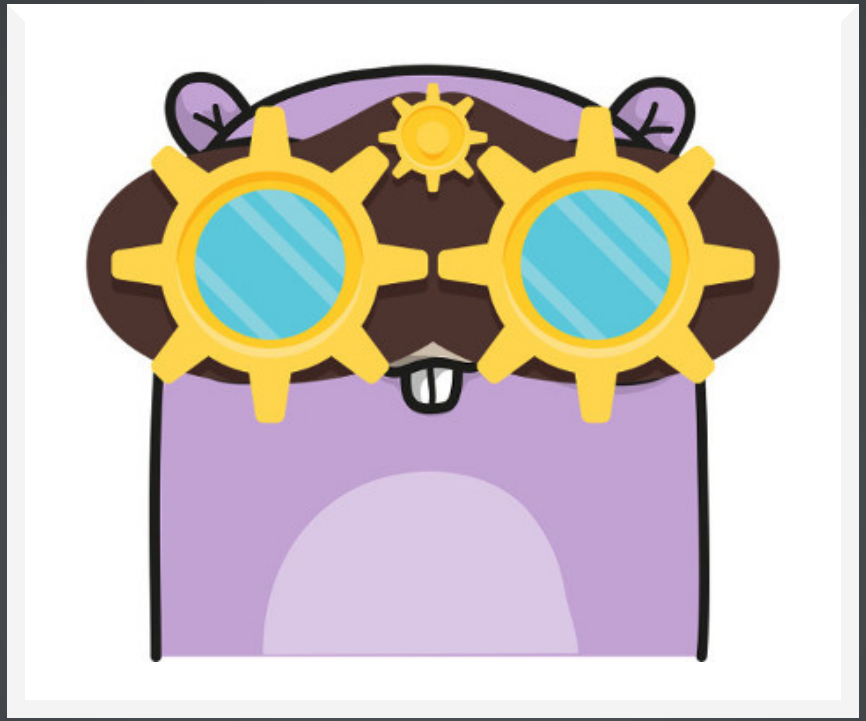


GOBOT.IO

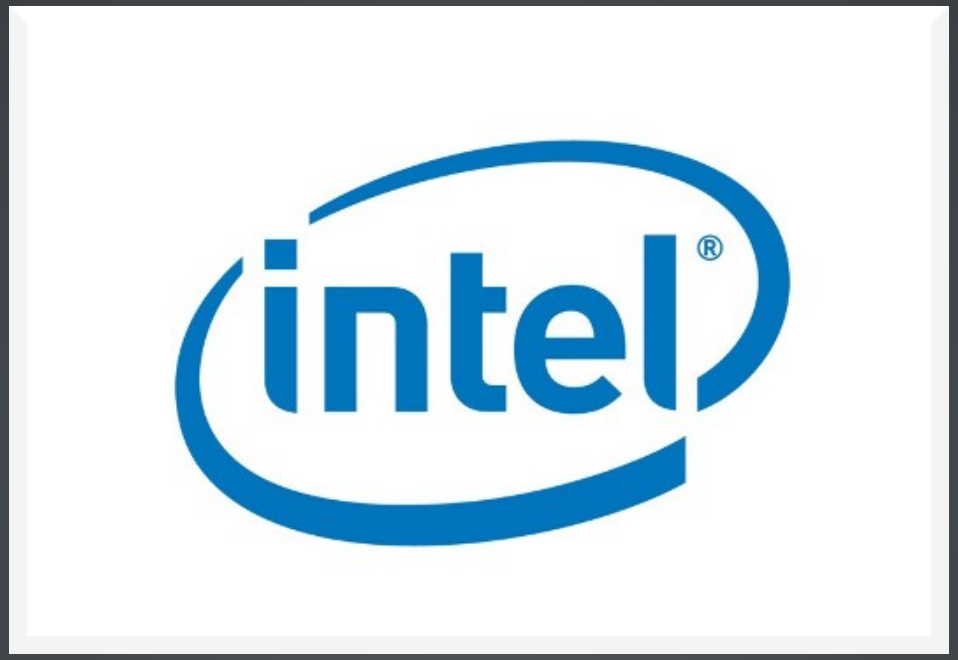


DJI.COM

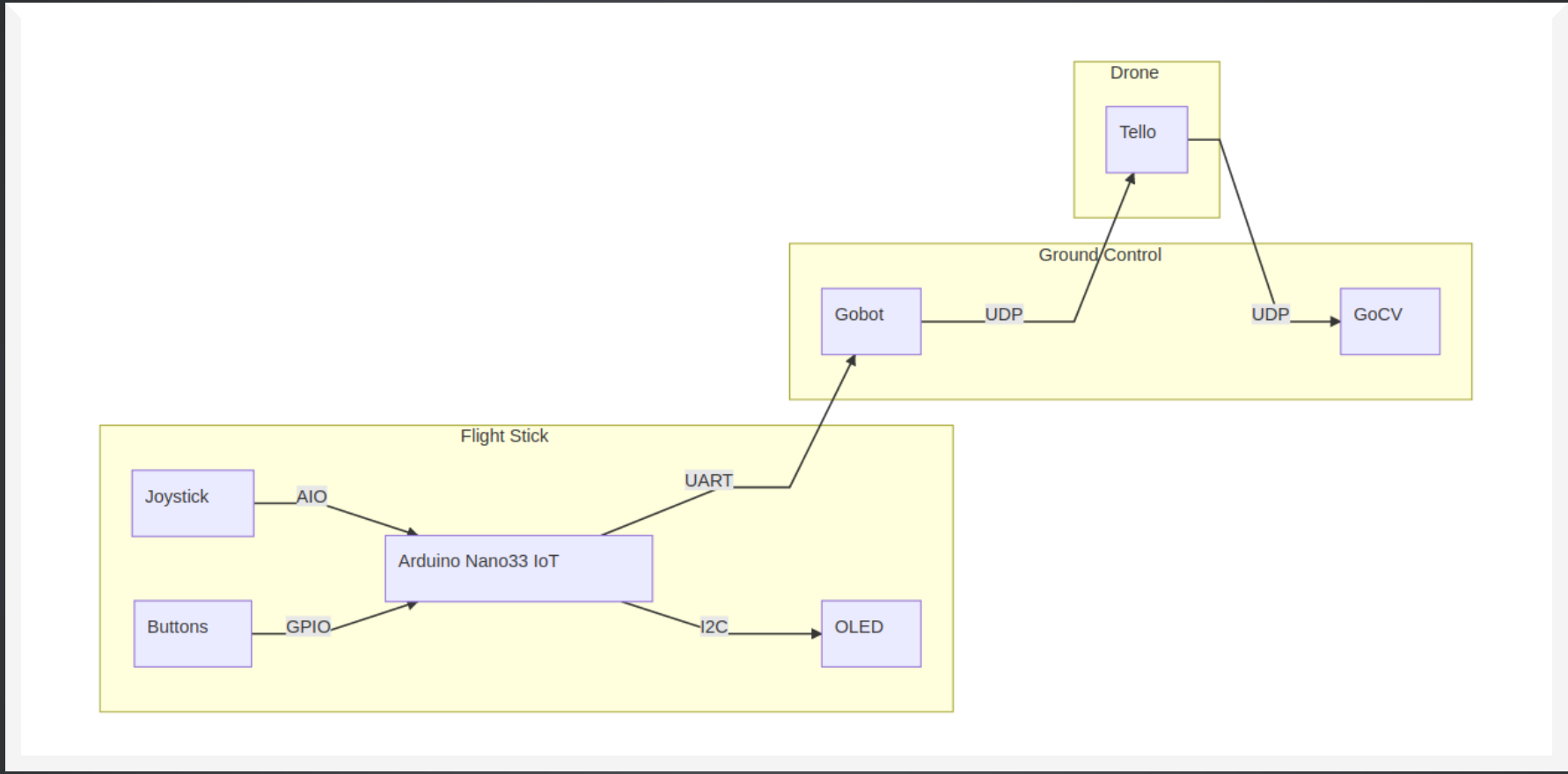
INTEL MOVIDIUS 2



GOCV.IO



INTEL OPENVINO



Flight Control System architecture

```
package main

import (
        "bufio"
        "fmt"
        "image"
        "image/color"
        "io"
        "math"
        "os"
        "os/exec"
        "strconv"
        "strings"
        "sync/atomic"
        "time"

        serial "go.bug.st/serial.v1"
        "gobot.io/x/gobot"
        "gobot.io/x/gobot/platforms/dji/tello"
        "gocv.io/x/gocv"
)

type pair struct {
        x int
        y int
}

const (
        frameX    = 400
```

DEMO

THE FUTURE OF EDGE COMPUTING

WEBASSEMBLY

TINYGO PLAYGROUND

PLAY.TINYGO.ORG

DEMO

RISC-V

SIFIVE HIFIVE1 REV. B

FREEDOM E310

RISC-V

32-BIT PROCESSOR

320 MHZ

4MB FLASH

DEMO

THE FUTURE IS HERE NOW.

TINYGO.ORG

HARDWARE HACK DAY
TOMORROW 9AM

THANK YOU!

@DEADPROGRAM

TECHNOLOGIST FOR HIRE