

TinyGo & micro:bit



Bärner Go Meetup – 28.11.2022

Whois



Lucas Bremgartner

Go Engineer for hire (contractor)

Open source enthusiast

Active in the Go community since ~8 years

Co-Organizer of [Bärner Go Meetup](#)

Online:

github.com/bremi | [@_breml_](https://twitter.com/_breml_) | [linkedin.com/in/lucas-bremgartner/](https://www.linkedin.com/in/lucas-bremgartner/)

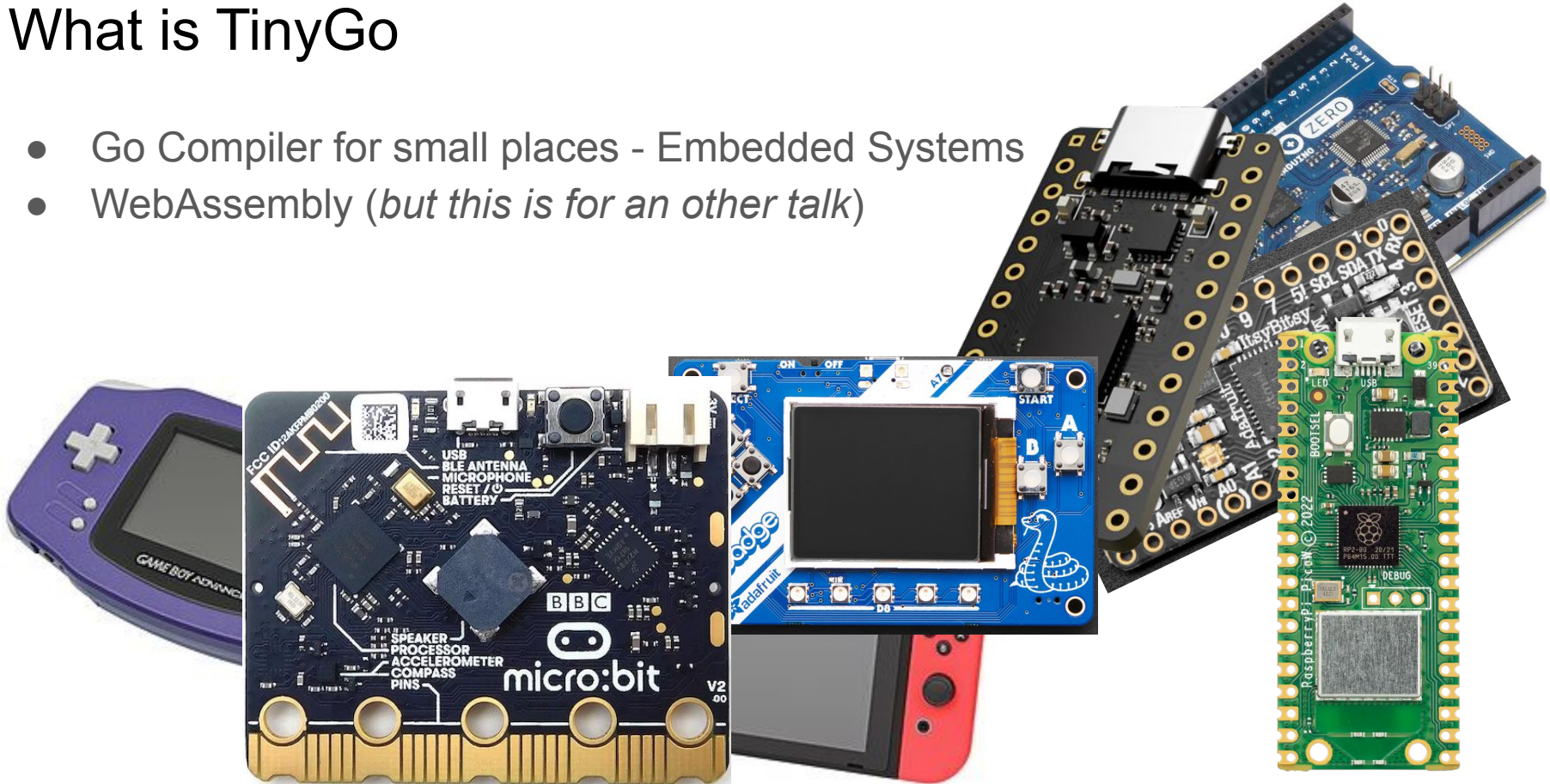


Agenda

1. What is TinyGo
2. What is the BBC micro:bit
3. Demo
4. Conclusions

What is TinyGo

- Go Compiler for small places - Embedded Systems
- WebAssembly (*but this is for an other talk*)



TinyGo Facts



[Ayke van Laethem](#) (NL)

First commit: April, 11th 2018, [Hello World with LLVM](#)

v0.1 (initial release) February, 1st 2019

Today (v0.26.0): > 3000 Commits, 122 Contributors, ~ 12k stars

85 different microcontroller boards supported

WASM for web browsers, server and edge computing environments,
WebAssembly System Interface (WASI) support

Since February 2020, [TinyGo is officially a Google sponsored project](#)

Web: <https://tinygo.org/> | Github: <https://github.com/tinygo-org/tinygo> | Playground: <https://play.tinygo.org/>

Differences between Go and TinyGo

TinyGo provides its own **runtime**, with support for bare metal (without OS)

Goroutines implemented as LLVM coroutines in a `async/await` model

Conservative Garbage Collection (`-gc=none|leaking|conservative`)

Pointers are determined by values that are in the range of valid pointer addresses:

```
ptr >= heapStart && ptr < heapEnd
```

provides a “**machine**” package for low level interactions with embedded systems

- Interrupts
- Volatile keyword (= memory mapped registers)

How does TinyGo work?



Sources: <https://tinygo.org/docs/concepts/compiler-internals/pipeline/> | <https://aykevl.nl/2019/04/llvm-from-go> | <https://getstream.io/blog/how-a-go-program-compiles-down-to-machine-code/> | <https://blog.gopheracademy.com/advent-2018/llvm-ir-and-go/>

Go vs TinyGo

Feature	Go	TinyGo
Basic types / Flow control / Slice	✓ / ✓ / ✓	✓ / ✓ / ✓
Interfaces / Closures / defer / recover	✓ / ✓ / ✓ / ✓	(✓) / ✓ / (✓) / ✗
Cgo / Reflection / Maps	✓ / ✓ / ✓	(✓) / (✓) / (✓)
Standard Library	✓	(✓)
Goroutines / Channels / Generics	✓ / ✓ / ✓	✓ / ✓ / ✓
Garbage Collector	✓	✓
Interrupts / Volatile (memory mapped registers)	✗ / ✗	✓ / ✓
Bare metal support / machine package	✗ / ✗	✓ / ✓

Sources: <https://tinygo.org/docs/reference/lang-support/> | <https://tinygo.org/docs/reference/lang-support/stdlib/> | <https://tinygo.org/docs/concepts/compiler-internals/interrupts/> | <https://tinygo.org/docs/concepts/compiler-internals/volatile/>

TinyGo Packages & stdlib

crypto/rand
device (subpackage per arch: arm, arm64, avr, esp, nrf, nxp, riscv, sam)
machine
machine/usb
net
os
os/exec
reflect
runtime
runtime/cgo
runtime/debug
runtime/interrupt
runtime/trace
runtime/volatile
sync
syscall
testing

From the official stdlib:

Total packages: 146

Importable: 126

Passes tests: 58

Sources: <https://github.com/tinygo-org/tinygo/tree/release/src> | <https://tinygo.org/docs/reference/lang-support/stdlib/> (Nov 2022)

Memory & Garbage Collection

heap blocks:

memory available for the garbage collector, in $4 * \text{size}(\text{ptr})$ blocks (e.g. 16 byte on 32 arch)

heap metadata

state of each heap block, one of: free, head, mark, tail

.bss:

global variables, entirely zero (optimization for simpler initialization)

.data:

global variables, not entirely zero

stack:

call stack, at the bottom of memory, grows from top to bottom, provides hardware checking for overflows



Source: <https://aykevl.nl/2020/09/gc-tinygo>

What is tiny



What is the BBC micro:bit

Designed by BBC (British Broadcasting Corporation) for computer education of 11 to 12 year old scholars in the UK.

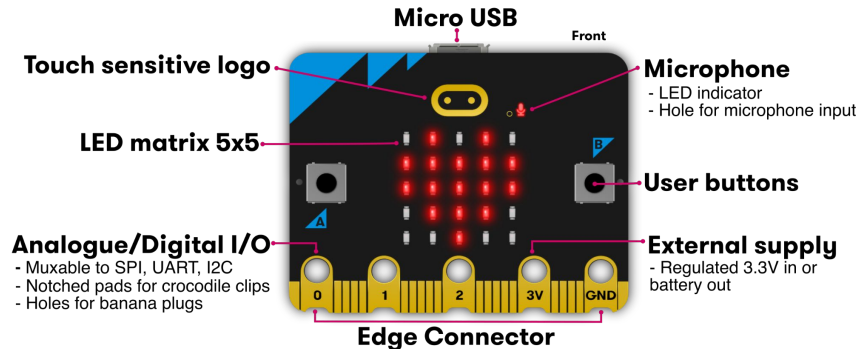
Price per unit ~ 25 CHF

V1

ARM Cortex-M0 32 bit CPU, 16MHz

Flash Rom: 256KB

RAM 16KB

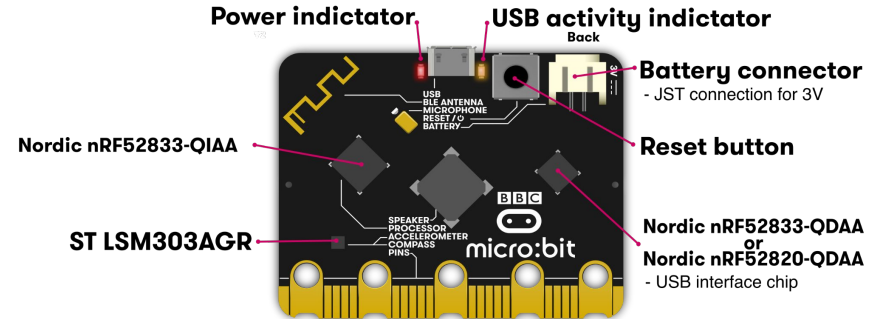


V2

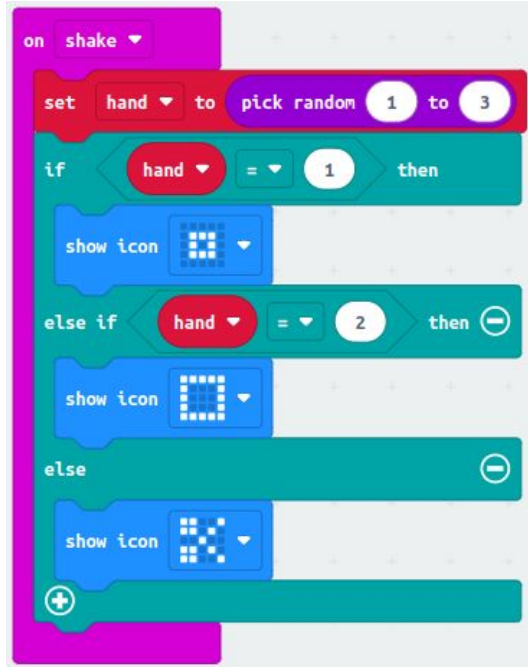
CPU: ARM Cortex-M4 32 bit with FPU, 64 MHz

Flash ROM: 512KB

RAM: 128KB



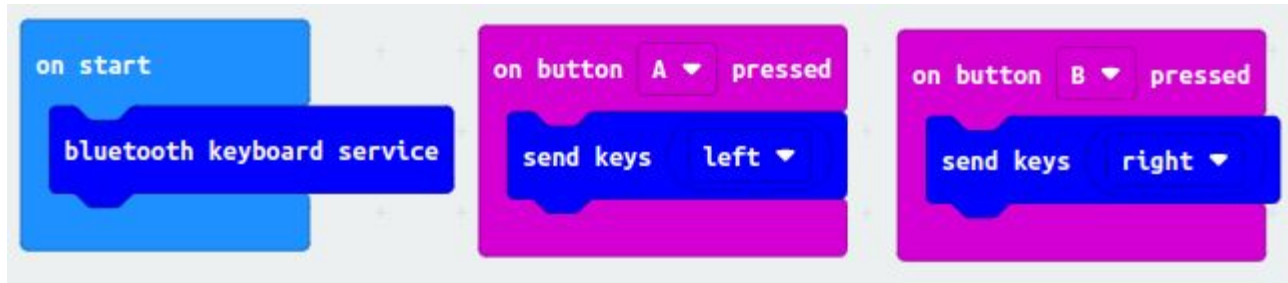
MakeCode and MicroPython



```
hand = 0
```

```
def on_gesture_shake():  
    global hand  
    hand = randint(1, 3)  
    if hand == 1:  
        basic.show_icon(IconNames.SMALL_SQUARE)  
    elif hand == 2:  
        basic.show_icon(IconNames.SQUARE)  
    else:  
        basic.show_icon(IconNames.SCISSORS)  
input.on_gesture(Gesture.SHAKE, on_gesture_shake)
```

More fancy: Bluetooth HID



TinyGo on micro:bit



TinyGo

```
package main

import ...

func main() {
    accl, _ := accelerometer.New()
    display := display.New()

    for {
        if accl.CurrentGesture() ==
            accelerometer.GestureShake {
            onShake()
        }

        time.Sleep(20 * time.Millisecond)
    }
}
```

```
func onShake() {
    hand := rand.Intn(3)
    switch hand {
    case 0: // Rock
        display.Show(image.SquareSmall)
    case 1: // Paper
        display.Show(image.Square)
    default: // Scissors
        display.Show(image.Scissors)
    }

    time.Sleep(1 * time.Second)
    display.Clear()
}
```


Conclusion

TinyGo:

- For my experiments, TinyGo is stable enough and covered everything I needed.
- The TinyGo VS Code extension and the VS Code serial console are of great help.
- For simple debugging use the serial connection, there is also support for GDB.

TinyGo for BBC micro:bit:

- Support:
 - OK: GPIO (General purpose IO), UART (serial), SPI (serial peripheral interface), I2C (I2C bus)
 - Partly: Bluetooth
 - Not yet: ADC (Analog digital converter), PWM (pulse width modulation), USB Device (storage)
- Much more low-level than compared to MakeCode or MicroPython
→ **but this also provides lots of opportunities**

Interfaces

Interface	Hardware Supported	TinyGo Support
GPIO	YES	YES
UART	YES	YES
SPI	YES	YES
I2C	YES	YES
ADC	YES	Not yet
PWM	Software support	Not yet
USBDevice	NO	NO
Bluetooth	YES	YES

Pull Requests while preparing this talk

- [tinygo-org/drivers #462 - microbitmatrix: fix inverted axis](#)
- [tinygo-org/drivers #472 - microbitmatrix: add brightness levels for LED pixels](#)
- [tinygo-org/drivers #478 - lsm303agr: fix I2C address auto increment for multi data read](#)
- [microbit-foundation/dev-docs #388 - Add TinyGo as other programming language with microbit support](#)
- [tinygo-org/tinygo-zoo #6 - Change alpha channel to non-transparent](#)
- [tinygo-org/drivers #486 - buzzer: tone duration float64, no tone during rest](#)

Questions



Thank you



Links

[TinyGo](#) | [TinyGo Playground](#) | [TinyGo VS Code Extension](#)

[Micro:bit hardware](#)

Technical details by Ayke van Laëthem:

- [LLVM from a Go perspective](#)
- [Garbage collection in TinyGo](#)
- [TinyGo Preview - how does it work?](#)

[GoCV](#) (used to capture the webcam)