

EMBD.

Golang Embedded Programming Framework



Star

434

Kunal Powar

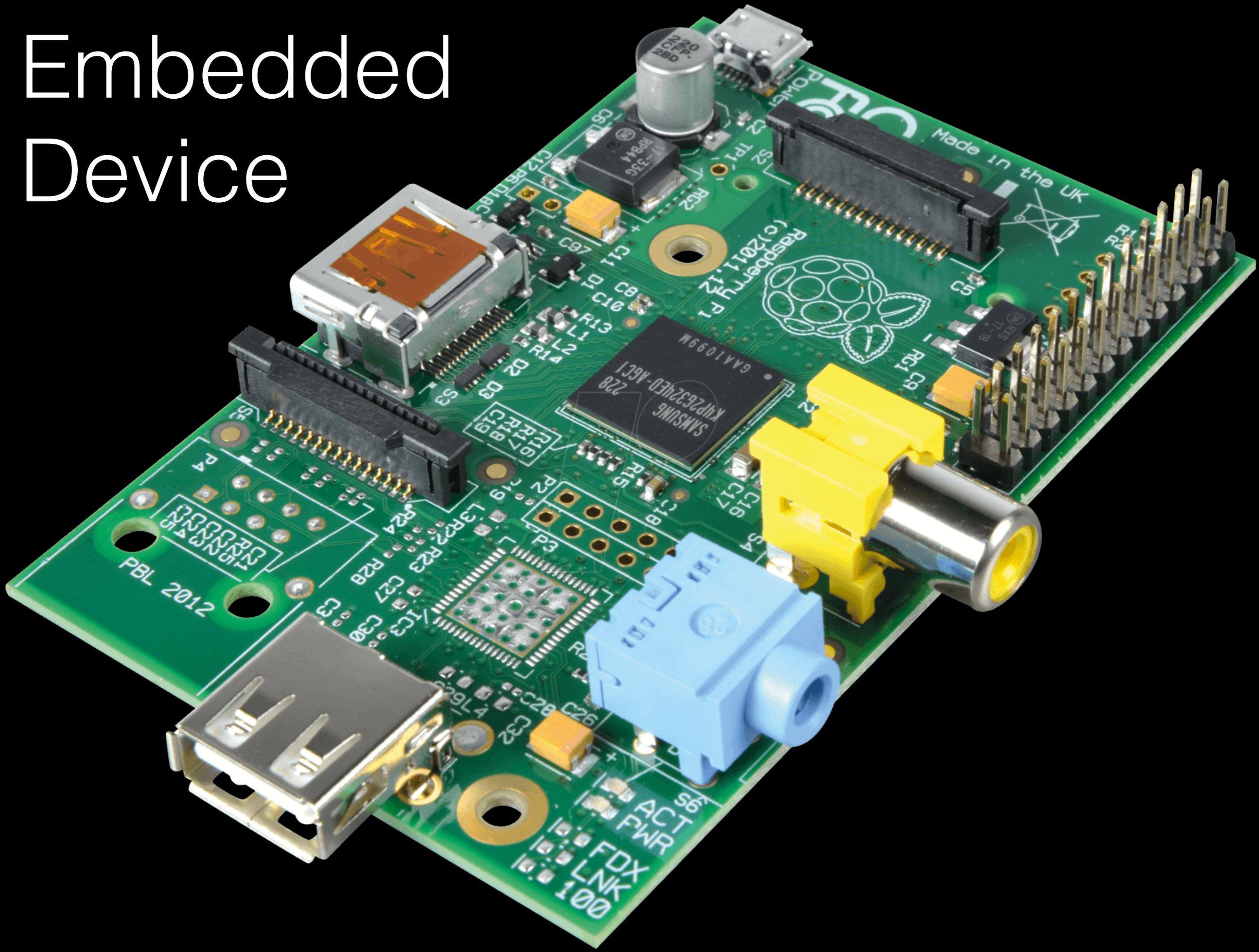


powar.io
github.com/kunalpowar
[@_poWar_](https://twitter.com/_poWar_)

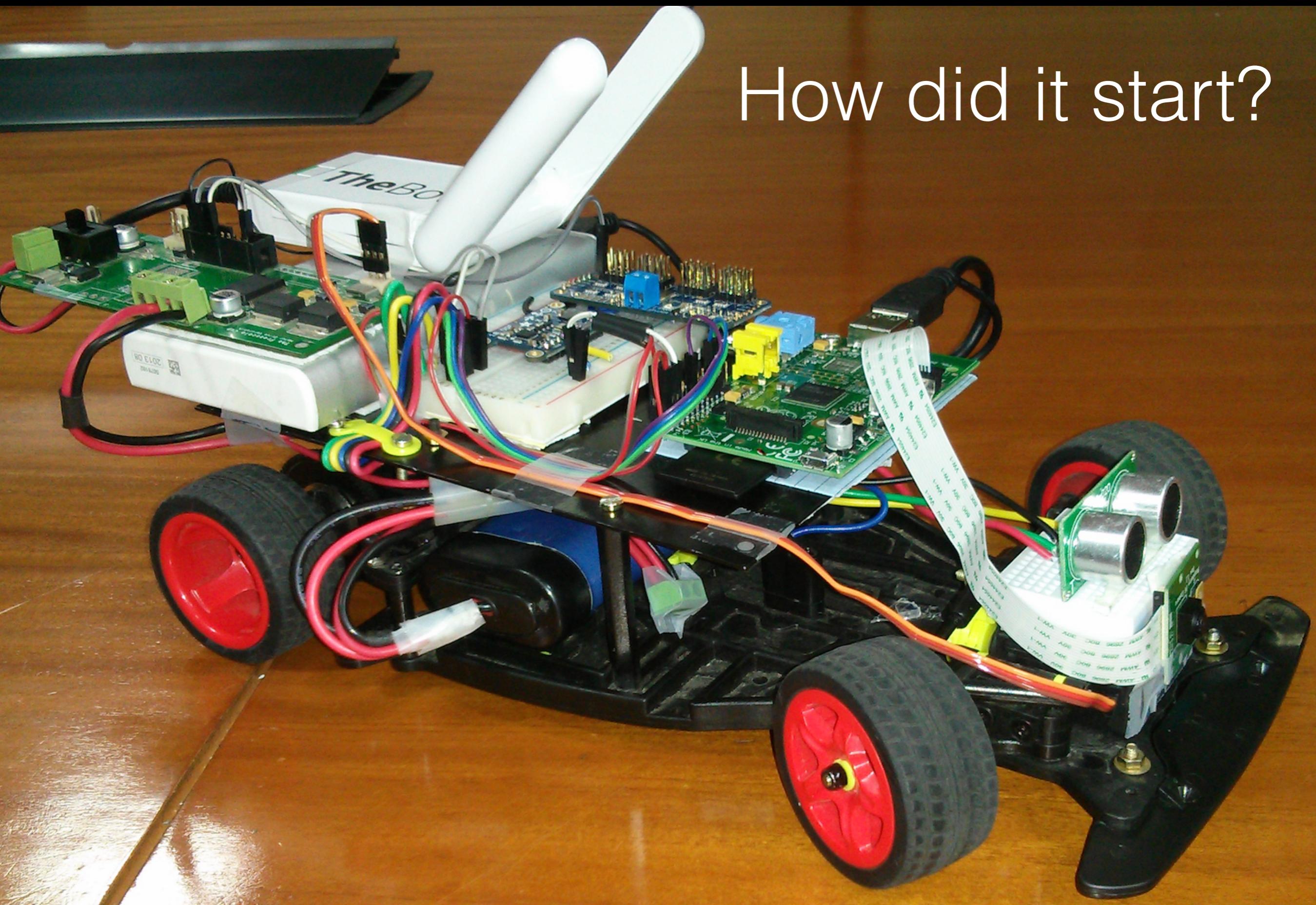


Embedded Programming?

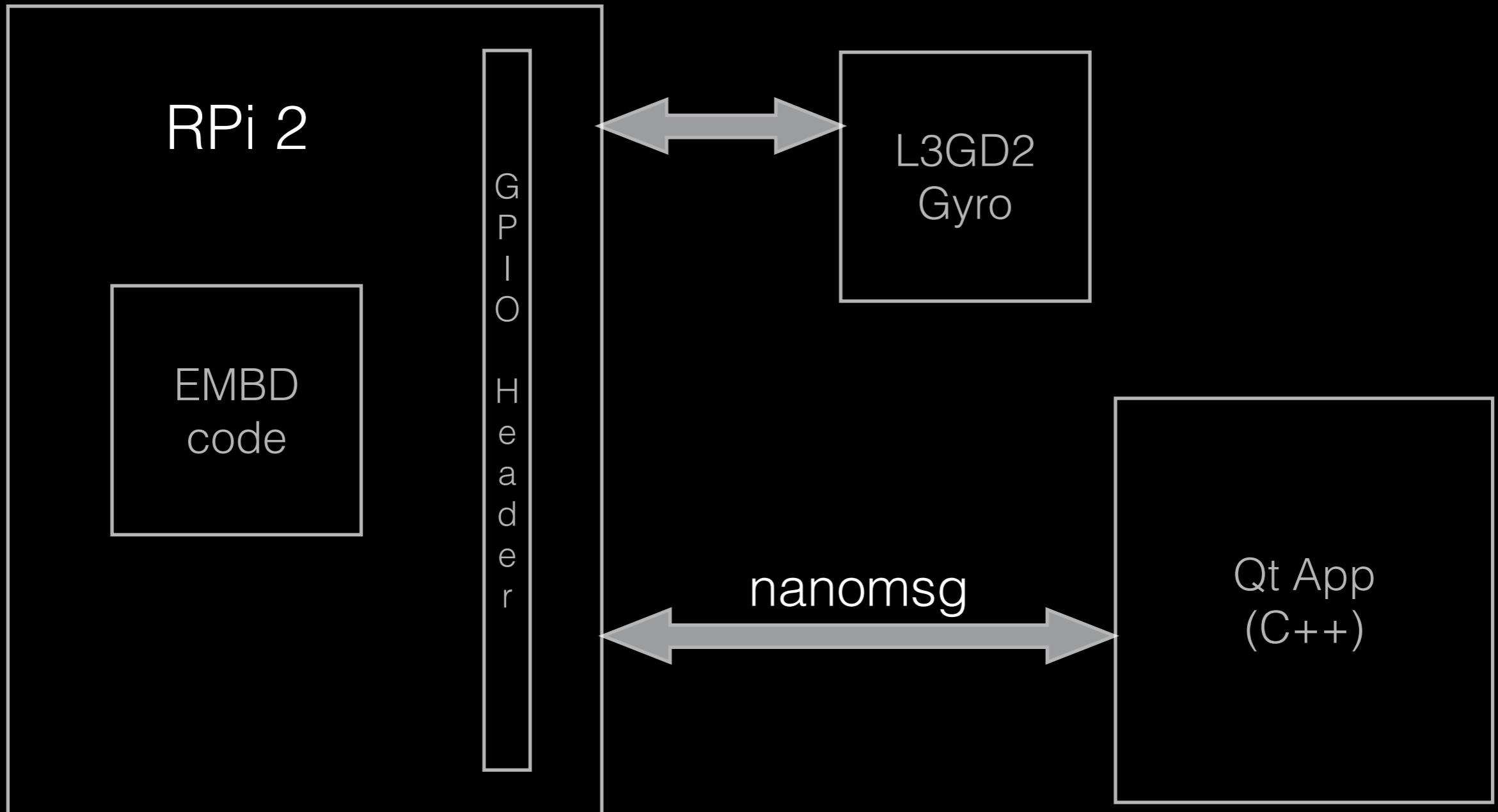
Embedded Device



How did it start?



Demo



Why and How
did we
use Go?

Why?

Easy to learn and use

Growing Community of gophers

Cross platform

Static binary

Lots of features which made sense in Hardware realm

Interfaces

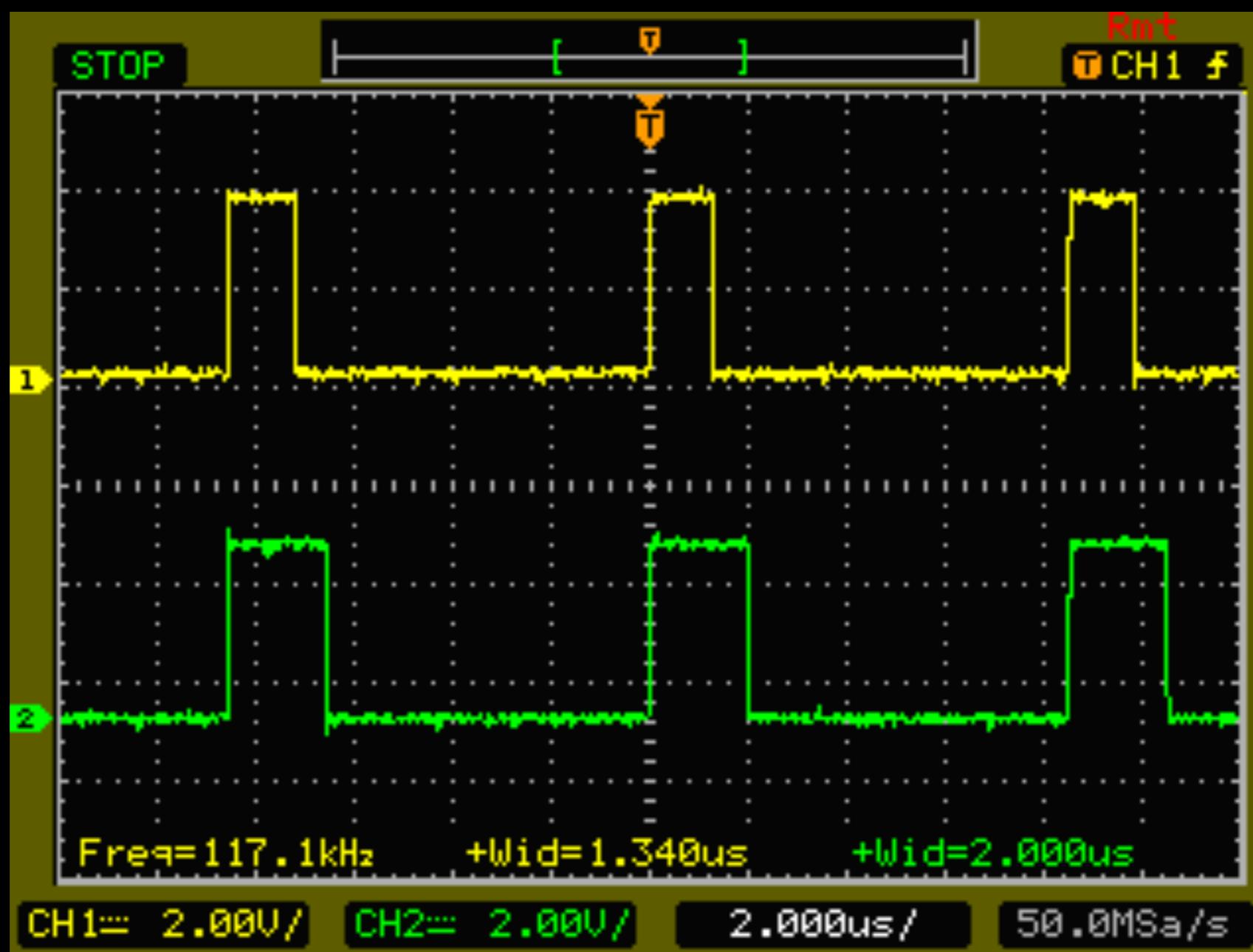
Servo

Can set the shaft to turn to a specific angle using a PWM signal as input.



PWM Signal

Pulse Width Modulated signal is a square wave, whose high/low duration can be changed.



```
type PWM interface {  
    SetMicroseconds(us int) error  
}
```

```
type Servo struct {  
    PWM PWM  
    ...  
}
```

```
func (s *Servo) SetAngle(angle int) error {  
    ...  
}
```

GPIO PWM (using ServoBlaster)

```
if err := embed.InitGPIO(); err != nil {  
    panic(err)  
}  
defer embed.CloseGPIO()
```

```
pwm, err := embed.NewPWMPin("P9_14")  
if err != nil {  
    panic(err)  
}
```

PWM from pca9685

```
if err := embed.InitI2C(); err != nil {  
    panic(err)  
}  
defer embed.CloseI2C()
```

```
bus := embed.NewI2CBus(1)
```

```
d := pca9685.New(bus, 0x41)  
d.Freq = 50  
defer d.Close()
```

```
pwm := d.ServoChannel(0)
```

Servo Usage

```
servo := servo.New(pwm)
```

```
servo.SetAngle(90)  
servo.SetAngle(10)  
servo.SetAngle(120)
```

```
defer func() {  
    servo.SetAngle(90)  
}()
```

Channels And Goroutines

```
type L3GD20 struct {  
    ...  
    orientations chan Orientation  
}  
  
func (s *L3GD20) Run() error {  
    ...  
    go func() {  
        // Sensor aquisition code.  
        ...  
        s.orientations <- someData  
    }()  
}
```

```
func (s *L3GD20) Orientations() (<-chan Orientation, error) {  
    ...  
    return s.orientations, nil  
}
```

```
if err := embed.InitI2C(); err != nil {  
    panic(err)  
}  
defer embed.CloseI2C()
```

```
bus := embed.NewI2CBus(1)
```

```
gyro := l3gd20.New(bus, l3gd20.R250DPS)  
defer gyro.Close()
```

```
// *
gyro.Run()

orientations, _ := gyro.Orientations()

for {
    select {
        case orientation := <-orientations:
            // Some logic
            ...
    }
}
```

* Pardon the lack of error handling

Now

3 of the most commonly used platforms (RPi 1 & 2, BBB)

6 of the most frequently used protocols (GPIO, I2C etc.)

12 of the most essential sensors (accel, gyro, etc.)

Journey Ahead

Support more platforms (Intel Edison etc.) and Firmata

Add driver code for more sensors

Cover all hardware communication protocols



Thank You

Kunal Powar

@_poWar_

github.com/kidoman/embed

github.com/kunalpowar/gophercon-2015

embed.io

youtu.be/iMXjkZ4B3EM (video of TheBot)



SOSTRONK.
HOME OF E-SPORTS

We are hiring :)