Optimization

Intro

- copy main.c to your project
- Check the code together
- NOTE: remind me to save code after each change we make please :)

What can we do better?

- Where do we want to start?
- What is the bottleneck?
- **Lot of things might in reality by optimized using compiler**
- Unfortunately for us, the C optimizations are turned off ($$$)
- Usually, it is better to start with the longest things

Always measure

- Premature optimization is root of all evil (KISS)
- Don't just guess, be sure what causes the problem
- Luckily, our architecture is pretty simple
- We can also inspect the generated assembly
- For the time sake, we will not do that here xD

Our options to measure

- Limited debugging options
- We can work just with simple led blinking
- Repeating procedure multiple times gives more precise results

1. Function calls

- First we can inline all the code to exactly see, what is happening
- Calling a function can require some work with registers
- There is enough device program memory that we can use

2. Repetitive operations

- DRY
- Quick operation * many repetitions = slow operation :)
- There might be some things we do repeatedly, can you spot them?
- HINT: check variables we are using repeatedly
- HINT: check functions we call repeatedly

3. Modulo operation is quite complex

- We can implement modulo in many different ways
- HINT: use overflow
- HINT: cut lower bits

Problem

- There is a bug in our algorithm
- Theoretically it should function correctly, but there are hidden delays
- Can you spot it?
- HINT: What color do we see when we set all LEDs to 0 except a single one?

4. Change algorithm

- Sometimes you just have to rewrite it :)
- We might be able to ease up constraints
- Do we need all 255 values for different colors for our use case
- We might care only about the color
- Any ideas how to change our algorithm?
- HINT: Discretize PWM to separate intervals with the same size.


That's all Folks!