Design decisions. Chose a balance between software scalability and system performance. Software to add more features but hardware to avoid system being overwhelmed by logic algorithms

* Did not implement hardware crc because stm32f042k6 does not come with programmable polynomial crc hardware. It can only do 32 bit crc, not 16. However, stm32f4 comes with it. Stm32L4 does not. For scalability, software was chosen even if less efficient. <https://community.st.com/s/question/0D50X00009XkYcq/calculate-16bit-crc-using-stm32f0-peripheral>
* I chose to use automatic baud rate for uart. It does not come with stm32f4 but does come with stm32l4(AN4908)
* [**https://medium.com/@charlesdobson/how-to-implement-a-simple-circular-buffer-in-c-34b7e945d30e**](https://medium.com/@charlesdobson/how-to-implement-a-simple-circular-buffer-in-c-34b7e945d30e) **circular bufefr**
* [**https://www.embedded.com/ring-buffer-basics/**](https://www.embedded.com/ring-buffer-basics/)
* **Since pwm works outside of cpu, perform multiple checks once pwm is triggered**
* **Didn’t feel the need to put a flag to check if string transmit to pc is compete or not as 10 sec very long**
* **Crc implementation** [**http://www.sunshine2k.de/articles/coding/crc/understanding\_crc.html#ch4**](http://www.sunshine2k.de/articles/coding/crc/understanding_crc.html#ch4)
* **Chose not to use mode for efficiency (https://embeddedgurus.com/stack-overflow/2011/02/efficient-c-tip-13-use-the-modulus-operator-with-caution/)**

**Chose not to use dynamic buffer and fixed as over time might corrupt the heap**