1. int main (void)
   1. reset\_wdt (); //wird nicht benötigt
   2. button\_polling ();
      1. <Hal/Button/inc/button.h>
         1. <Hal/Gpios/inc/pin.h> 🡪 **Anpassung auf cyhal**
   3. demo\_control ();
      1. (SW 3 pressed🡪 switch context 0….4 ) “democontrol.h”

MINIMAL\_RGBI, 0

XYZ\_TO\_FLASH,

COLOR\_CORRECTION,

RUNNING\_LIGHT,

UPDATE\_RATE\_EXAMPLE,

UART\_MODE 5

* + 1. Switch Case 🡪 check 100ms flag (🡪 hier ein SW Timer mit 100ms Tick oder FreeRTOS?)
       1. minimal\_rgb\_stripe\_control ();
          1. init\_led\_feature\_blocking (true, false, &countLed, &init); //init-BiDir

hal\_reset\_osire\_start (); (leere function, tbd ob gebraucht wird)

errorCodeLed = osp\_reset (0🡪broadcast an alle);

hal\_reset\_osire\_end (); (leere function, tbd ob gebraucht wird)

**enum** **OSP\_ERROR\_CODE** **osp\_reset** (**uint16\_t** deviceAddress) osireDevice.c

ospErrorCode **=** **osp\_cmd\_buffer** (**&**ospCmd)**;** 🡪 Error Paremeter Handling und build header (ganzen RX Stream)+response ja/nein+rx lenght

spiError **=** **send\_data\_over\_spi\_blocking** (ospCmd**.**p\_outCmdBuffer**,**ospCmd**.**outCmdBufferLength)**;**

**enum** **OSP\_ERROR\_CODE** **osp\_osire\_set\_pwm** (**uint16\_t** deviceAddress**,osirePwmData\_t** data) osireDevice.c

**enum** **OSP\_ERROR\_CODE** **osp\_osire\_read\_pwm** (**uint16\_t** deviceAddress**,osirePwmData\_t** **\***p\_rsp) osireDevice.c

* + - 1. color\_Correction\_Stripe\_Control (colorCorrectionModeWithTemp);
      2. save\_xyz\_in\_flash (one\_time\_button\_pressed\_sw2 ());
      3. running\_light\_control (runningLightVersion);
      4. update\_example\_control ();
  1. uart\_receive\_new\_msg ();

NEXT: TODO: