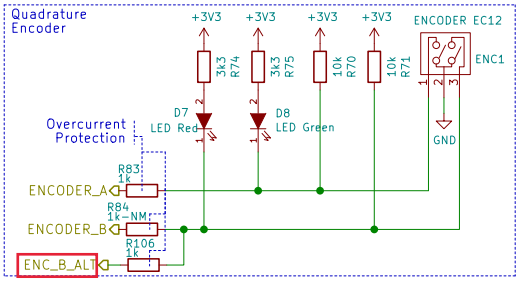

Figure 5.2: Encoder section schematic on PMDB16

5.1.1 Configuring the GPIOs

As seen in chapter 3, to configure a GPIO whe must

- Reset the peripheral
- Set the peripheral clock for the port
- Set the GPIO Registers
  - Configure P6 and P7 as Inputs
  - No need to set output type or output speed (since they are input)
  - Set Pull-up /pull down to none (we have external pull-ups -see schematics-)

5.1.1.1 Peripheral reset

Looking at the Reference manual[1], we see that the reset register to use is RCC_AHB1RSTR.

In the file stm32F4xx.h we see the definition for that register and the #define for the constant we can use[2].

So, to reset the peripheral we would do:

CODE

```
RCC->AHB1RSTR|=RCC_AHB1RSTR_GPIOCRST; //Reset peripheral
RCC->AHB1RSTR&=~RCC_AHB1RSTR_GPIOCRST;//Run the peripheral
```

This approach can be error prone, since the constant used are defined as #defines in the .h code. A much better way to proceed should re-define the RCC type using unions and bit fields, in order to be able to reset the peripheral with instructions like

CODE

```
RCCb->AHB1RSTR.GPIOCRST=1; //Reset the peripheral
RCCb->AHB1RSTR.GPIOCRST=0; //Run the peripheral
```

that is much more human-readable, using also the same names for the bits used in the data-sheet. To be able to do so we can copy the definition of RCC_TypeDef from the .h file and create a union between the 32 bit value of the register and a bitfield structure, changing the line

[1] Search for “peripheral reset” on the pdf file, or go to section 2.3.4-5-6 of the reference manual

[2] Search for “RCC” in the file stm32F4xx.h, located under CMSIS->device in our project