The purpose of this section is to discuss some of the challenges and benefits that is related to the use of an FPGA in this project. The FPGA connects the microcontroller all the components on the pan-tilt system in order to be able to control the system. It collects the data from encoders and homing sensors, and it creates and updates the PWM signal to each of the H-Bridges.

refrace a bit

One of the main differences between an FPGA and a microcontroller is that on an FPGA everything is implemented in hardware. This also means that all of the processes running, is running in parallel. When implementing the different components this needs to be taken into consideration.

The FPGA used in this project is a Artix7 mounted on a BASYS3 kit.

ref

0.0.1 System overview

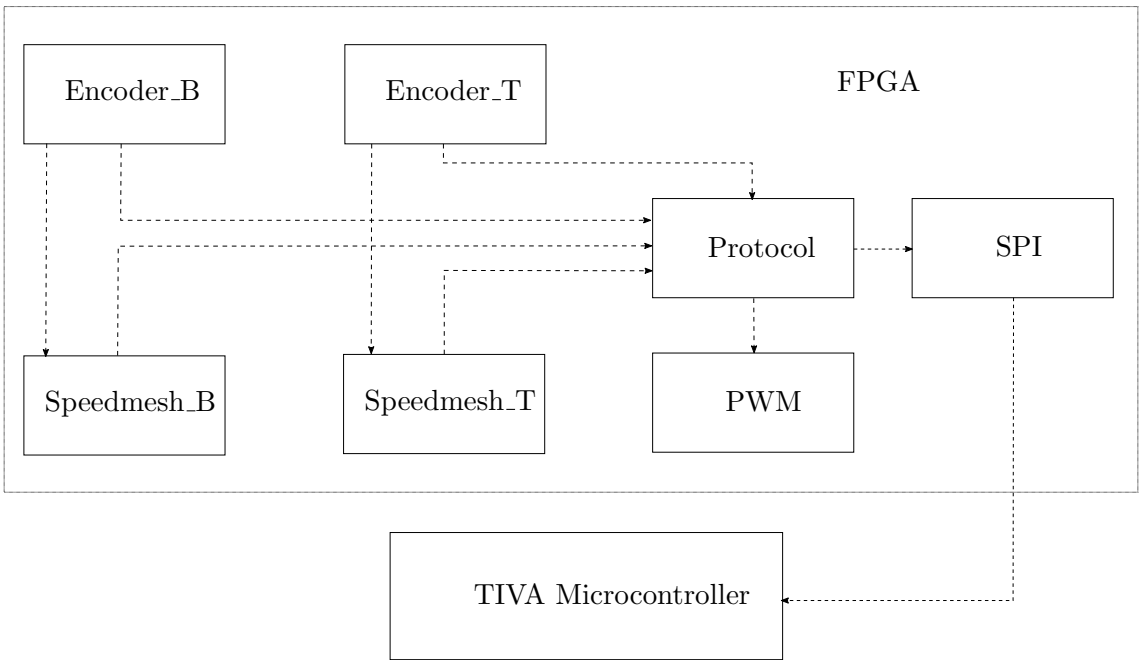

Figure 1: *System overview.*

The system consist of different modules, implemented in VHDL. To store the data that needs to be exchanged between the FPGA and the Tiva microcontroller, different registers is created with the necessary size. Se table 1.

The connection between the modules can be seen in the system overview. Se figure 1

12 Bit	9 Bit	1 Bit
Position_T	PWM_T	Home_T
Position_B	PWM_B	Home_B
Velocity_T	-	Reset_T
Velocity_B	-	Reset_B

Table 1: *FPGA Registers*

0.0.2 PWM

The PWM module creates a PWM signal for both motors. It takes a 9 bit vector as input for each motor, and output the signals necessary for the H-Bridges to work. It takes the 100Mhz system clock from the FPGA as a input, and then the module contains a clock divider that creates the necessary clock frequency, for the desired PWM frequency. MSB in the 9 bit input vector determine witch way the motor should turn. The module is design so that it is possible to just define the system clock frequency and the desired PWM frequency, and the correct divider is calculated when the code is synthesized. This is done so that it is easy to change the PWM frequency for testing.

0.0.3 Protocol

The protocol module uses a SPI module created by for communicating with the Tiva ref. microcontroller. The role of the protocol module is to decode the data received over SPI, and prepare data for transmission, in regards to the previously determined protocol.
Since all the different processes, like receiving data over SPI and updating data registers, all run in parallel, it is important to make sure that the protocol module cannot update the data registers while the SPI module is receiving data form the microcontroller. This is done by placing a latch on the output from the protocol module, witch is triggered by a busy signal from the SPI module.

0.0.4 Quadrature decoder

0.0.5 Speed measurement