FPGA (Emil, semi-Morten & semi-Jonathan)

* PWM signal
  + Construct PWM signal
* Feedback of motors
  + Reset encoder to match base position
    - Make decoder to read encoded message
    - Read from hall sensors
* SPI coms to Micro controller
  + Send messages from FPGA to microcontroller
  + Receive messages from microcontroller

Micro controller (Jonathan, Morten & Pascal)

* Regulation (Jonathan & Pascal)
  + Construct current controller
    - Sense current
      * Use sen pins with a restistor and read voltage to get current
  + Construct velocity controller
    - Sense voltage
      * Use voltage combined with equation in lecture 3 in regulation slide 34
    - Calculate speed
  + Construct position controller
    - Use encoder data for position controller
  + Setup regulation system
    - Find the constants
    - Make controllers connected
* SPI coms to FPGA & PC (Morten)
  + Get PC data
    - Connect PC pos data to pos controller
  + Get FPGA data
  + Send data to FPGA
  + Send data to PC (Debugging)
* Setup priority scheduling (Morten)

Vision (Aksel & Mads)

* Find ball
  + Maybe use flooding to find ball easier [see link](http://www.roborealm.com/tutorial/Marble_Maze/slide010.php)
* Find line
  + Use OpenCV example from [stack overflow](https://stackoverflow.com/questions/45322630/how-to-detect-lines-in-opencv)

Design of components (Ellen & Pascal)

* Design board
  + Get aluminum board
  + Drill holes in edges to fasten to robot
  + Drill holes for camera holder
* Design camera holder
  + Get camera from Leon (Maybe)
  + Construct camera holder
* Design counter weight
  + Match camera holder design
  + Match weight of camera