Ultrasonic measure

# 12.09.16 Kokkuvõte

1. Projekti juhitakse nõudmiste “requirements” kaudu.
2. Projekti elluviimiseks kasutame MCU platvormi Texas Instrumentsist.
3. MCU – mikrokontroller, mikroskeemi sees on CPU, mälu, perifeeria.
4. Projekt sisaldab põhifunktsionaalsust ja muud funktsionaalsust (nt. diagnostika)
5. Esiteks teeme põhifunktsionaalsust valmis.
6. Projekt hakkab kasutama kihilist disaini (hierarchy)
7. Esimesel kihil on riistvalarilne MCU moodul (nt. UART, GPIO, timer, etc.)
8. Järgmisel kihili on tootja (TI) “driver library”
9. Järgmine kiht on meie poolt tehtud perifeeria draiver, mis kasutab tootja draiverit ja eraldab riistvara osa funktsionaalsest koodist. See tähendab, et kood, mida hakkab meie dariverit kasutama, ei kasuta enam mitte midagi riistvara spetsiifilist.
10. Järgmisena tulevad seadme “device” draiveri kood. Ehk kood, mis tegeleb seadmetega väljaspool MCUt, mis on ühendatud MCU perifeeria kaudu. Kood kasutab selleks eelmises punktis valmiskirjutatud perifeeria draiverit.
11. Selle üle võib veel minna applikatsiooni kood, mis tegeleb konkreetse funktsionaalsuse implementatsiooniga. Väga võimalik, et ka meie projekti puhul võib selline kood tekkida.

# Ülesanned:

1. Code Composer Studio (CCS) install.
2. Git programmi allalaadimine ja install.
3. TI Driverlib allalaadimine TI veebilehelt.
4. GitHub konto registreerimine ja projekti “checkout” oma arvutisse. Mind registreerida kasutajana.
5. Üks tiimiliige:
   1. Lugeda MCU manuaalist kõike infot UART perifeeria kohta. Meid huvitab: töö põhimõtted, mida on vaja teha, et tööd alustada, lugeda üks bait, kirjutada üks bait, teada puhvri staatust?
   2. Arusaada Driverlib näitekoodist UARTi kohta. Mõelda kuidas saaks seda tööle katkestusi kasutamata.
6. Teine tiimiliige:
   1. Otsida infot LCD displei ja tema UART ülemiku kohta Sparkfun kodulehelt [www.sparkfun.com](http://www.sparkfun.com)
   2. Meid huvitab: töö põhimõtted, mida tuleb initisialiseerimiseks teha, kuidas kuvada ja kustutada?
7. Lisaülesanne(kui kellegil oleks aega):
   1. Lugeda MCU manuaalist kõike infot GPIO (general purpose I/O) perifeeria kohta. Meid huvitab: töö põhimõtted, kuidas sättida konkreetsele pinnile vajalik funktsioon, sättida konkreetse pinni sisendiks/väljundiks, sättida pinnile digitaalne väärtus 1 või 0?
   2. Arusaada Driverlib näitekoodist UARTi kohta. Vaadata kuidas saab katkestusi kasutada.