Projekt i Inbyggda system och signaler

Provkod 1506 HT16-VT17

Syftet med examinationsuppgiften är att studenten skall tillämpa kunskaper, färdigheter och värderingsförmåga från kursen för att lösa ett problem inom området inbyggda system och signaler.

Planering och deadlines

Studenterna får arbeta ensam eller i grupper upp till maximalt fyra personer för att lösa uppgiften. Om ni arbetar tillsammans ska ni dela upp problemen och i den tekniska rapporten ange vem som var ansvarig för vilka problem. Alla gruppmedlemmar ska ändå vara insatta i alla delar.
Alla delar i rapporten ska vara framtagna av gruppmedlemmarna själva och beskrivna med egna ord, ni får alltså inte plagiera någon annans lösning eller text.
I samband med slutredovisningen kommer examinatorn att ställa muntliga förståelsefrågor till enskilda studenter för att försäkra sig att alla studenter har uppnått lärandemålen.

Måndag 19 december

- Rob och Christoffer kommer och presenterar projektuppgiften
  - Demo och exempel på API mot Axis kameror

- Utlåning av utrustning
- Möjlighet till frågor

Tisdag 20 december

- Gruppindelning
  - Uppgiften görs i grupper upp till fyra studenter, välj själva vilka som vill jobba tillsammans
  - Alla grupper ska ha skickat namn och github-id på vilka som ingår i respektive grupp till ulrik.eklund@mah.se senast tisdag kväll

Onsdag 18 januari

- Rob eller Christoffer kommer och svarar på frågor
- Milstolpe som bör vara klar:
  - Utkast på problembeskrivning
  - Utkast till lösning: "napkin diagram" på systemlösning
  - Val och installation av utvecklingsmiljö klart
  - Kod som snurrar på kortet

    - Kontakt med kameran via webb-API
    - git repo etablerat

Söndag 19 februari

- Inlämning av rapport på Itslearning
- Länk till kodrepo på github

Onsdag 22 februari

- Demonstration, troligtvis på Axis i Lund
    - Förutsatt att den inlämnade rapporten är tillräckligt bra!

Söndag 26 februari

- Möjlighet att uppdatera rapporten innan slutlig bedömning
    - baserat på studenternas egna erfarenheter och feedback från demon

- Slutlig uppdatering av git-repo

Inlämningar

- Teknisk rapport på itslearning
    - Det är viktigare att bedriva hur problemet och lösningen ser ut än vilka aktiviteter studenterna har gjort
    - Tips om hur man skriver en teknisk rapport från KTH - *använd checklistan!*

- Kod, tester och testresultat i git-repo med länk
- Muntlig presentation, inklusive svara på frågor och visa förmåga att förklara och resonera kring problem och lösning

Utrustning

- Axis-kamera
- Adafruit Huzzah utvecklingskort ESP8266: *"ESP8266 WiFi Starter Kit is designed for people who want to use the new low-cost ESP8266 WiFi microcontroller and are comfortable with that platform to explore IoT. An ESP8266 development board is included. An Arduino or other microcontroller is not required. This pack DOES NOT include an FTDI cable or USB console cable - you will need one to program & power the HUZZAH ESP8226. A full size breadboard and some prototyping wires are also recommended. Some soldering is required to assemble the HUZZAH ESP8266 and HDC1008 sensor."*
- FTDI-cable
- Adafruit Huzzah utvecklingskit för ESP8266

Programmering och verktyg

Det är starkt rekommenderat att använda sin egen dator i kursen. All nödvändig programvara kommer att finns tillgänglig för installation.

Korten med ESP8266 går att programmera i olika miljöer, det är upp till varje grupp att välja och installera lämpligast utvecklingsmiljö för ert problem och lösning, se https://learn.adafruit.com/adafruit-huzzah-esp8266-breakout/downloads.
Oavsett språk så skall givetvis koden var välstrukturerad, prydlig och välkommenterad.

- Lua, baserat på NodeMCU, finns redan flashat på Adafruit Huzzah-kortet
    - Fördelar: Kräver minimal installation, firmware finns redan installerat. Viss förmåga att köra parallella processer. Stöd för http
    - Nackdelar: Inget enkelt stöd för schemaläggning och tidsstyrning. Nytt språk som måste läras.
- Arduino IDE for ESP8266, github
    - Fördelar: Enkelt att programmera om man kan C. Visst stöd för tidsstyrning.
    - Nackdelar: Behöver flasha om firmware på kortet. Inget RTOS
- microPython
    - Fördelar: Lätt att hantera http. Funkar på flera olika host-PC
    - Nackdelar: Behöver flasha om firmware på kortet. Inget enkelt stöd för schemaläggning och tidsstyrning. Nytt språk som måste läras
- Espressif RTOS, GitHub
    - Fördelar: Bygger på C. Använder sig av FreeRTOS. Bibliotek för http. Många kodexempel.
    - Nackdelar: Behöver flasha om firmware på kortet. Byggmiljön är bara Linux-baserad

All kod skall lämnas in som en länk repo på GitHub. Varje grupp kommer att få ett eget privat repo om de vill. Rapporten bör innehålla en beskrivning av hur repot är organiserat om det inte är uppenbart.
De som går i trean rekommenderar gitkraken som git klient som har bra stöd för flera utvecklare att jobba tillsamman i samma repo.

Bedömning

Projektuppgiften kommer att bedömas inom tre huvudområden:

1. **Hur väl lösningen löser "problemet".** Bedömningen baseras i huvudsak på den tekniska rapporten, och hur väl den tar upp nedanstående punkter (bedömningen kompletteras även av den muntliga presentationen). Detta relaterar till följande lärandemål

    1. självständigt konstruera och testa programvara för inbyggda system innefattande parallella aktiviteter, digitala filter och regleralgoritmer
    2. integrera olika delar av ett inbyggt system till ett fungerande autonomt system

2. **Kvaliteten på implementationen.** Bedömningen baseras i huvudsak på systemdiagram tillsammans den den inlämnade koden (via GitHub). Bedömningen tar också hänsyn till hur väl koden och rapporten stämmer överens. Detta relaterar till följande lärandemål:

    1. funktion och programmering av avancerade mikroprocessorsystem
    2. realtidsaspekter i samband med inbyggda system 3. användning av typiska sensorer i mobila enheter och signalbehandling med mikroprocessor

3. **Planering, genomförande, dokumentation och presentation**, detta relaterar till följande lärandemål

    1. använda moderna utvecklingshjälpmedel för programutveckling för inbyggda system
    2. genomföra uppgifter inom givna tidsramar
    3. kritiskt förhålla sig till innehåll i litteratur, teknisk dokumentation, marknadsföringsmaterial och material på internet
    4. visa förmåga att söka, analysera och diskutera vetenskapliga artiklar om tekniska aspekter beträffande utveckling och programmering av inbyggda system samt deras roll i samhället och om människors ansvar för hur de används

Betygsättning

För betyg 3 krävs

- Problemet är tydligt definierat i både text och bilder. Om det är säkerhetsrelaterat finns det en analys av den fysiska placeringen och eventuella risker
- Lösningen löser problemet.
- Det är beskrivet vem som drar nytta av lösningen och vilket värde den har för den intressenten.
- Systemlösningen är beskriven i både text och diagram
    - Beskrivningen inkluderar både lösningskomponenter, fysiska komponenter och tillhörande programvara.
    - Beskrivningen är fullständig och innehåller alla delar, även de som inte utvecklats av teamet.

- Systemlösningen har validerats att den löser just problemet, t.ex. genom testning. Testfallen och resultaten är beskrivna.
- Det finns en beskrivning hur systemet ska användas.
- Systemet fungerar vid demonstrationen.
- VAPIX API är korrekt implementerat.
- Signalanpassning och filtrering har gjorts av sensorer.
    - Sensorerna är anpassade och ansluten till mikroprocessorn på ett korrekt sätt. In- och utgångar är korrekt definierade.
    - Det finns beskrivningar och diagram över hur alla konstanter och parametrar har identifierats och

kalibrerats.

- Schemaläggning och tidsstyrning av processer är beskriven på ett korrekt sätt i rapporten.
- Studenterna har valt och använt utvecklingsmiljö och dokumenterat detta.
- Rapport är inlämnad innan deadline och systemet demonstreras vid schemalagt tillfälle.
- Studenten har använt sig av litteratur, teknisk dokumentation och material på internet som refereras och korrekta hänvisningar görs i den tekniska rapporten.
- Rapporten beskriver vem som har gjort vad (om man inte jobbat ensam)
- Rapporten är skriven på svenska eller engelska, på ett begripligt språk med fullständiga, sammanhängande meningar.

För betyg 4 krävs utöver kraven för betyg 3:

- Problemet är relevant och verkligt.
- Eventuella etiska aspekter tas upp i rapporten om det är relevant
- Det finns beskrivning både för användning och "installation"
- Realtidsaspekter beaktas i programmet och programmeringen har gjorts med preemptive tasks i RTOS eller "round-robin" som kör olika processer för olika uppgifter. Det finns en diskussion om val av tidskontanter i den tekniska rapporten.
- Koden följer en väldefinierad kodningsstandard, antingen en egen (inkluderas i repot) eller en extern (med referens).
- Studenterna kan använda sig av utvecklingsmiljön för att leta fel i programvaran.
- Det finns genomförda enhetstester för olika delar av programvaran.

För betyg 5 krävs utöver kraven för betyg 3 och 4:

- Lösningen är nyskapande
- Problemet har lösts utan att utrustning gått sönder under projektets gång.
- Lösningen kan hantera felfall (vilka dokumenteras i rapporten)
- Koden skall vara väl strukturerad med tydligt och logisk uppdelning i funktioner och källkodsfiler. Kopplingar mellan olika delar av koden skall ske på konsekvent sätt.
- Implementationen är effektiv och följer "best practice" för ett web-API.
- Schemaläggningen av processerna beräknas att den fungerar (enligt t.ex. Rate-Monotonic-Analysis) och baseras på mätningar eller uppskattningar av koden.

Följande ger typiskt ett underkänt betyg:

- Problemet och dess beskrivning går inte att förstå
- Systemlösningen har ingen relevans till problemet
- Systemet är inte beskrivet i både diagram och text
- Ingen validering eller testning har gjorts av systemlösningen
- Det finns ingen manual eller användningsbeskrivning
- Systemet fungerar inte vid demonstrationen
- Koden är inte exekverbar på mikroprocessorsystemet.

- VAPIX API är inte implementerat så det fungerar korrekt
- Det finns inga mekanismer i koden som hanterar tidsaspekter eller styr exekveringen av programvara
- Studenten har tagit exempel från internet och/eller litteratur utan att ange källor eller förklara varför
- Matematiken och annan teori stämmer inte eller följer inte vedertagen teori inom ämnet
- Matematiken i rapporten stämmer inte överens med den kod som finns i projektet.
- Rapporten är inte inlämnad vid deadline
- Alla olika uppgifterna är inte lösta eller innehåller fel (signalbehandling, kommunikation, linjärisering, plottar, mm)
- Rapporten beskriver inte vem som har gjort vad (om man inte jobbat ensam)
- C-programmeringen följer inte vanliga konventioner och praxis
- Referenser görs inte till etablerade källor, som läroböcker eller tekniska manualer
- Det finns inte beskrivningar och diagram över hur alla konstanter och parametrar har identifierats.
- Schemaläggning av processer är inte beskriven eller beskriven på ett felaktigt sätt i rapporten.
- Det går inte att granska koden i projektet då den är ofullständig eller obegriplig.
- Språket i rapporten är inte förståeligt, därmed går det inte heller att bedöma innehållet.
- Rapport eller kod är plagierat från någon annan student.