

PROJECTE ADSTR

PROJECTE 18/19:

OBJECTIUS:

- Dissenyar i implementar una aplicació de control d'un sistema en una plataforma embedded per assolir les competències específiques.
- Desenvolupar les competències transversals establertes en l'assignatura d'Anàlisi i disseny d'un sistema de temps real.

1. CONTROLADOR EMBEDDED

El controlador que implementarem es desenvoluparà sobre un sistema de plataformes embedded (Arduino i Raspberry Pi)

El sistema manté una la temperatura propera a una consigna que li ha estat indicada. Per realitzar aquesta funció adquireix la temperatura amb un sensor i s'actua sobre un ventilador segons un determinat algorisme de control a través d'un relé.

2. PARTS DEL PROJECTE

A nivell de maquinari es disposa d'una placa Raspberry Pi, un sensor de temperatura LM35 i un Atmega328P (en la placa Arduino). i d'un relé.

Posteriorment s'afegiran altres elements a controlar de forma distribuïda i utilitzar diferents sistemes de comunicació (IoT).

El projecte es dividirà en 3 parts:

2.1 Part 1: Sistema de captura de la temperatura, control del sistema i registre de les alarmes.

- Cal crear un programa en codi C anomenat 'captura' que s'executarà des de RPi.
- Aquest **llegeix cada minut** la temperatura del sensor LM35. Aquesta funció es realitzarà a través d'un convertidor A/D de l'Atmega328P i s'aprofitarà el programa i el protocol de comunicacions desenvolupat a Informàtica Industrial el passat curs 2017-2018. Annex I (Protocol Control Sensor - Sistema Embedded)
- El programa controla un relé que acciona un ventilador via una sortida digital de l'Atmega 328P.
- El control és un algorisme simple: si la temperatura supera una consigna establerta s'activa el ventilador.
- Insereix la informació a la base de dades implementada sobre SQLite que es guarda a la RPi.
- El programa haurà de registrar a la base de dades si el sistema porta 5 minuts sense poder controlar la temperatura amb el ventilador en funcionament. Aquest registre s'ha de continuar fent cada 5 minuts en cas que la situació es mantingui.
- Posar-ho en marxa de forma automàtica quan s'inicia el sistema en la RPi.

-La base de dades ha de tenir la següent estructura:

-Taula temperatura

Camps:

-Data/hora

-Temperatura

-Estat ventilador (0: apagat, 1: posem en funcionament, 2: continuem en funcionament)

-Taula alarmes

Camps:

-Data/hora

-Temps funcionant el ventilador ON

En el desenvolupament de les anteriors funcionalitats caldrà satisfer els següent requeriments:

- El codi ha d'estar en un projecte a *GitHub* i cal portar un control de versions amb *GIT*
- Caldrà que el codi C segueixi el **conveni** d'estil establert en el projecte kernel Linux i la documentació del codi s'ha de fer en DOXYGEN.
- El projecte s'ha de compilar, netejar i instal·lar usant d'un fitxer *Makefile*
- Caldrà utilitzar les funcions temporals de la *librt* per controlar el període de mostratge en la captura de la temperatura.
- El programa permetrà passar per paràmetres per línia de comandes (p.ex. `nom_bbdd`)

2.2 Part 2: Informe diari per e-mail

El sistema haurà d'incorporar les següents funcionalitats afegides a les de la primera part:

- Cal crear un programa en codi C anomenat 'informe'.
- El programa consulta la base de dades implementada en SQLite, on realitzi un informe de l'evolució de la temperatura a les darreres 24 hores que ha anat guardant el programa 'captura'.

Aquest informe l'enviarà per e-mail i haurà de tenir la següent informació:

- Data/hora inici,
- Data/hora final,
- Temperatura màxima,
- Temperatura mínima,
- Temperatura mitjana,
- Temps total que el ventilador ha estat funcionant,
- Temps mitjà de funcionament del ventilador,
- Número de cops que ha funcionat el ventilador,
- Llistat d'alarmes: data/hora inici, temps ventilador ON

- Aquesta aplicació s'executarà periòdicament un cop cada 24 hores (cada dia) a través del servei *cron* del sistema operatiu.

- El programa permetrà passar per paràmetres per línia de comandes. (p.ex. `adreça_destí`)

- Posar-ho en marxa de forma automàtica quan s'inicia el sistema en la RPi.

- Cal crear una funció que enviarà el correu electrònic amb l'informe. Aquesta funció ha de compilar-se per separat i crear una biblioteca dinàmica.

- El programa principal que genera l'informe diari haurà d'utilitzar aquesta biblioteca per enviar correu.

- La resta del codi del programa de l'informe diari ha d'estar en dos fitxers font (.c): un amb la funció `main` i l'altre amb la resta de funcions que s'hagin creat.

També caldrà ampliar la Part 1 (programa 'captura') per tal de notificar alarmes per e-mail:

- El programa 'captura' haurà d'utilitzar la biblioteca anterior creada per enviar e-mails per notificar les alarmes. Aquesta funcionalitat nova al programa de la *Part 1* és per enviar un e-mail d'alarma quan el ventilador estigui actiu més de 5 minuts, indicant que el sistema té dificultats fer disminuir la temperatura en el moment d'enregistrar l'alarma a la base de dades. Cada 5 minuts es torna enviar e-mail si aquesta continua activa en el moment i es registra a la base de dades.

En el desenvolupament de les anteriors funcionalitats caldrà satisfer els següent requeriments:

- El codi ha d'estar en un projecte a *GITHUB* i cal portar un control de versions amb *GIT*
- Caldrà que el codi C segueixi el **conveni** d'estil establert en el projecte kernel Linux i la documentació del codi s'ha de fer en DOXYGEN.
- El projecte s'ha de compilar, netejar i instal·lar usant d'un fitxer *Makefile*

2.3 Part 3: Ampliació del sistema de forma distribuïda.

Es vol capturar altres dades de sensors per crear un sistema de monitorització distribuït, a cada grup s'especificarà els paràmetres a capturar.

S'utilitzaran diverses RPi per a aquesta finalitat.

Aquesta part s'implementarà amb llenguatge Python

La arquitectura distribuïda es dissenyarà amb cada grup.

En cada dispositiu cal muntar un servidor que pugui entregar les dades sota petició d'un client.

-El sistema servirà una pàgina web simple amb la informació del sistema i les dades registrades.
Cal acordar amb el professor quina estructura s'implementa per donar aquest servei.

3. ORGANITZACIÓ I TERMINIS

El projecte es realitzarà en grups de 2 estudiants.
A EUSSternet es publicarà la composició de cada grup.

Cada grup tindrà un secretari que realitzarà l'acta de seguiment del treball.
Aquest càrrec serà rotatori.

Periòdicament caldrà realitzar una reunió de 10 minuts amb presència del professor per revisar l'estat d'execució del projecte i planificar la feina fins a la següent setmana.

Els terminis són els següents:

- Part 1: 23-octubre-2018
- Part 2: 27-novembre-2018
- Part 3: 18-gener-2019