

# Pràctica 1: Compensador de llum

**Asignatura:** Sistemes de Temps Real

**Alumnes:** David Gutiérrez Pasqual

**Data d'entrega:** 23/12/2021

# Decisions de disseny

- Totes les tasques les he creat com a fils per tenir més control sobre la prioritat de cadascuna (baixa, normal o alta).
- El buzzer s'activa quan els sensors (llum, potenciòmetre i botó) tenen valor negatiu.
- Tasques (implementades com a mètodes) creades:
  - **lightReader**: s'encarrega de llegir el sensor de llum i el tractament del valor d'aquest.
    - Si el valor del sensor és negatiu llavors la variable **lightVal** no s'actualiza així que manté el mateix valor fins que el sensor torna a donar un resultat positiu.
  - **potReader**: lectura del potenciòmetre i tractament de la dada.
  - **buttonReader**: lectura del botó i crida del mètode `onButtonPress` si es compleix la condició (botó premut).
    - Per tal de limitar l'accés a la tasca **onButtonPress** he implementat la variable global **avgThread** perquè no es formi una cua si l'usuari prem el botó (és bastant sensible així que hi hauria diversos input a la vegada).
    - El valor de la condició per executar el fil de **onButtonPress** és 0.75 perquè és el valor aproximat que dona el botó quan es prem.
  - **valuesDisplay**: s'encarrega d'imprimir els valors tant periòdics (percentatges d'intensitat de la llum i d'intensitat del led) com aperiòdics (mitjana després de 10 segons) a la pantalla LCD.
    - Per tal de que la tasca `valuesDisplay` sàpiga quins valors ha de mostrar per pantalla he implementat la variable global **avgValue** que a la vegada s'utilitza per emmagatzemar el resultat del càlcul de la mitjana.
    - Utilitzo un semàfor perquè és un recurs compartit (valors de llum i led juntament amb el valor de la mitjana després de 10 segons).
  - **buttonPress**: s'encarrega de calcular la mitjana de valors de lluminositat capturada després de 10 segons.
    - Assigno la variable **avgThread** a true al principi del mètode i a fals al final perquè el programa sàpiga quan pot permetre l'accés de més input.

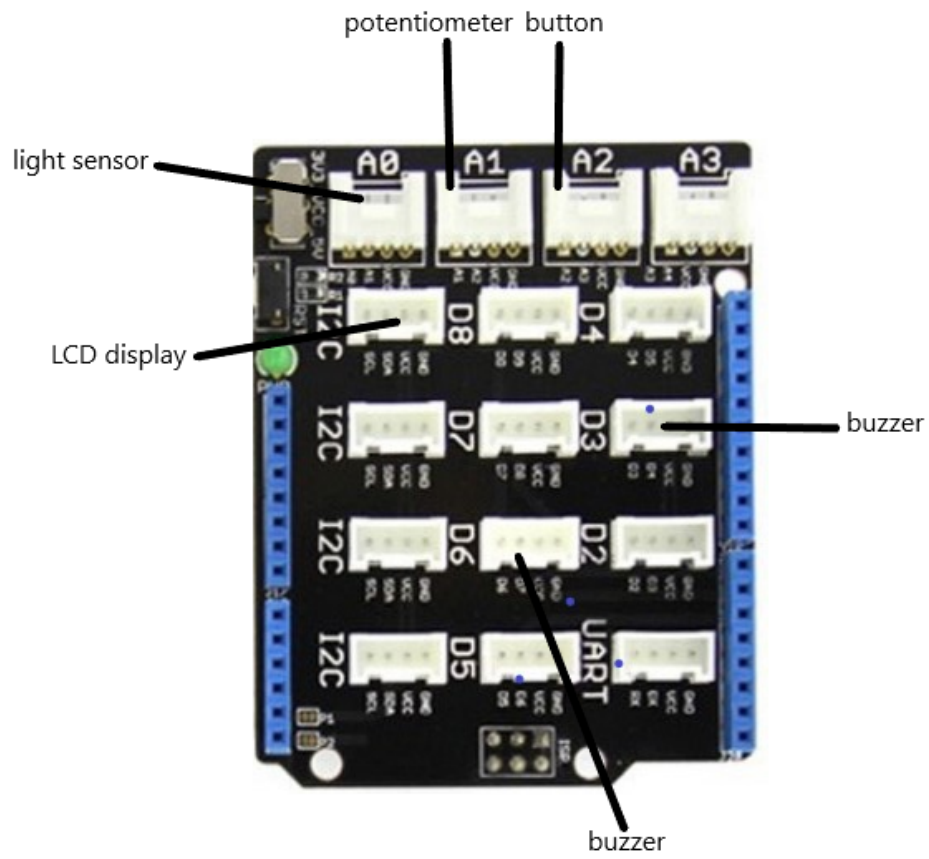
- **buzzerEmitter**: s'encarrega d'activar el buzzer durant 500 mil·lisegons.

## Planificació de les tasques

Definició de les prioritats i assignació de la periodicitat de cada tasca. Com ja s'ha pogut observar anteriorment les tasques periòdiques són aquelles que llegeixen valors contínuament.

- **lightReader**: és una tasca periòdica que s'executa en prioritat alta.
- **potReader**: és una tasca periòdica que s'executa en prioritat alta.
- **buttonReader**: és una tasca periòdica que s'executa en prioritat alta.
- **valuesDisplay**: és majoritàriament periòdica però també té una part periòdica, la de mostrar el valor de la mitjana després de 10 segons.
- **buttonPress**: és una tasca aperiòdica que s'executa en prioritat normal per no parar l'execució d'altres tasques de menor o igual prioritat.
- **buzzerEmitter**: és una tasca aperiòdica que s'executa en prioritat normal per no parar l'execució d'altres tasques de menor o igual prioritat.

## Esquema de connexionat



# Conclusions

Aquest treball m'ha servit per aprofundir més sobre les tasques en sistemes de temps real (tant les prioritats com les periodicitats) tot i que no he utilitzat totes les estructures que estaven a la meua disposició com **condition variables**, **event flags**, **mail** i **queue**.