Introducción

En nuestro proyecto, tenemos como objetivo, el poder brindar a la población una seguridad en el momento en el que ingieran agua, esta cumpla con los requerimientos de calidad para su consumo.
Se implementará el uso de 3 sensores para cuantificar el estado del agua y a través de un chip STM32F407VGT6 se realizará la lectura y envío de datos a través del protocolo de comunicación Lora, que será recibido por una Rapsberry, que a su vez lo subirá a la nube que nos permite recoger y almacenar datos de sensores y desarrollar aplicaciones IoT.

Justificación

Si se encuentra en un estado de procesamiento y supera un rango determinado entones eventualmente se activará una alarma.

```
ltl p_error1 {
        [ ] (((state == s_process) && [ ](range)) -> <> [ ](alert == 1))
}
```

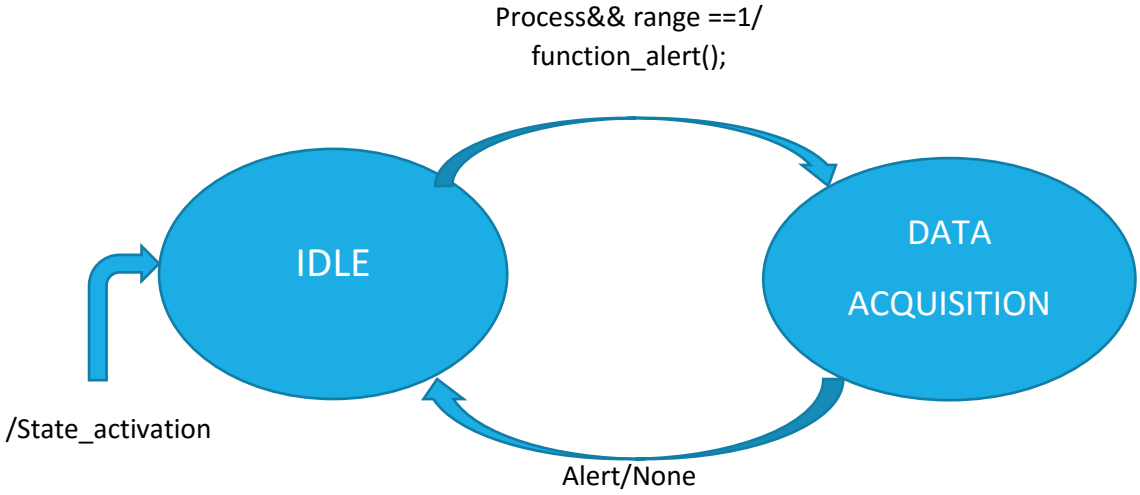
Si es encuentra en un estado de medida y se presenta un error determinado entones eventualmente se activará una alarma

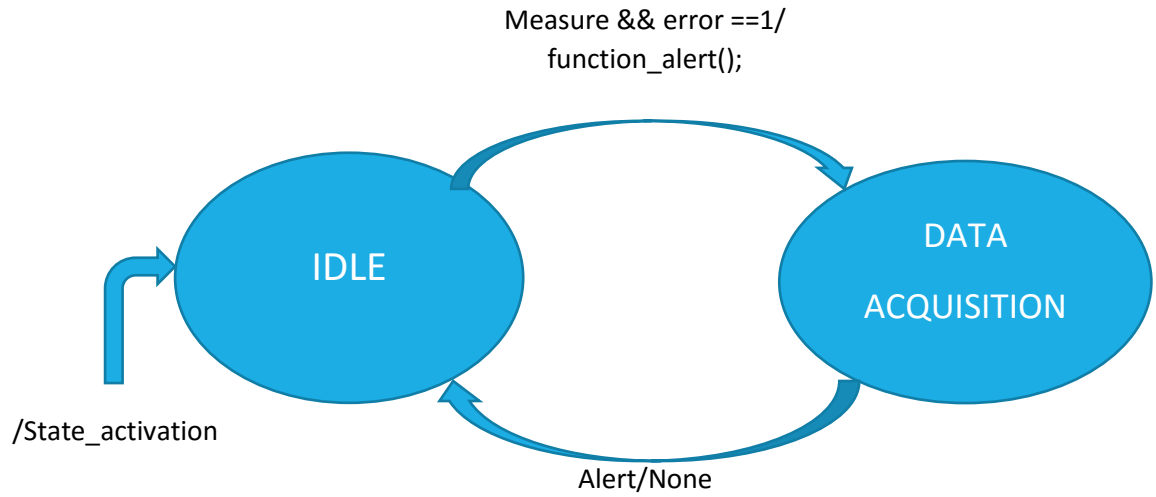
```
ltl p_error2 {
        [ ] (((state == s_measure) && [ ](error)) -> <> [ ](alert == 1))
}
```

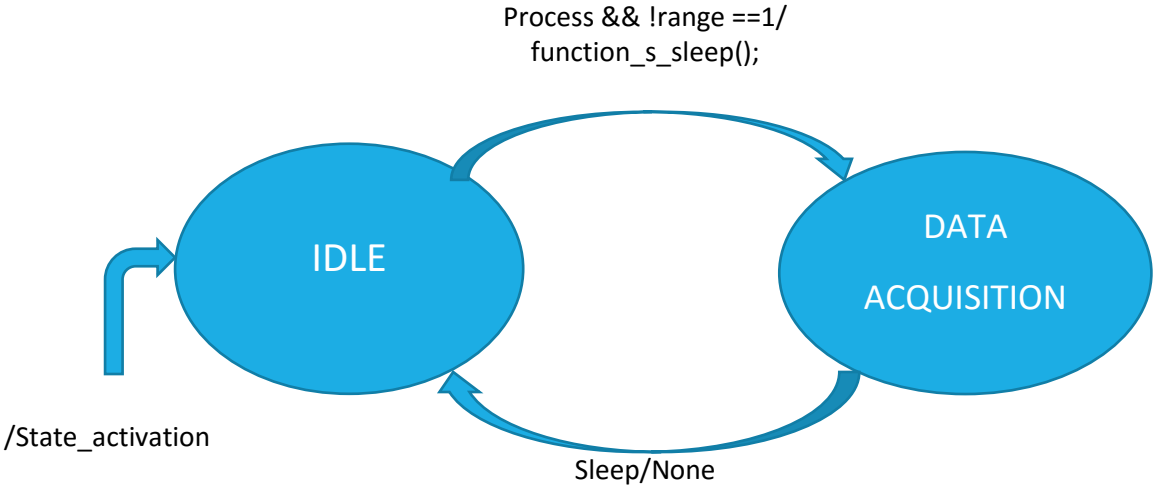
Si se encuentra en un estado de procesamiento y no se supera el rango determinado entonces eventualmente entra en un estado de dormir para lo que es ahorro de energía.

```
ltl p_error3{
        [ ] (((state == s_process) && [ ](!range)) -> <> (state == s_sleep))
}
```

Modelo de la Maquinas de estados







Resultados de la verificación con Spin.

Al realizar la comprobación del sistema, mediante la generación de un *active proctype*. Se inicializaron los estados para realizar la comprobación en Promela.

```
active proctype entorno () {
        if
                :: presence = 1;
                :: error = 1;
                :: range = 1;
                :: timer = 1;
                :: (!presence && !error && !range && !timer) -> skip
        fi
        printf("Presencia = %d, error = %d, alerta = %d \n", presence, error, alert);
}
```

Realizando la comprobación con el comando Spin -a LSEL.pml

```
D:\Spin>spin -a LSEL.pml
ltl p_error1: [] ((! (((state==3)) && ([] (range)))) || (<> ([] ((alert==1)))))
ltl p_error2: [] ((! (((state==2)) && ([] (error)))) || (<> ([] ((alert==1)))))
ltl p_error3: [] ((! (((state==3)) && ([] (! (range))))) || (<> ((state==0))))
  the model contains 3 never claims: p_error3, p_error2, p_error1
  only one claim is used in a verification run
  choose which one with ./pan -a -N name (defaults to -N p_error1)
  or use e.g.: spin -search -ltl p_error1 LSEL.pml
```

Ejecutamos el comando Spin -search – ltl p_error1 LSEL.pml

```
State-vector 36 byte, depth reached 18, errors: 0
       77 states, stored
       39 states, matched
      116 transitions (= stored+matched)
       13 atomic steps
hash conflicts:         0 (resolved)

Stats on memory usage (in Megabytes):
    0.004       equivalent memory usage for states (stored*(State-vector + overhead))
    0.281       actual memory usage for states
   64.000       memory used for hash table (-w24)
    0.343       memory used for DFS stack (-m10000)
   64.539       total actual memory usage


unreached in proctype sensor_fsm
        LSEL.pml:36, state 17, "state = 2"
        LSEL.pml:42, state 25, "state = 4"
        LSEL.pml:43, state 28, "state = 3"
        LSEL.pml:40, state 31, "printf('Estado : Measure \n')"
        LSEL.pml:49, state 36, "state = 4"
        LSEL.pml:47, state 41, "printf('Estado : Process \n')"
        LSEL.pml:54, state 45, "alert = 1"
        LSEL.pml:56, state 48, "alert = 0"
        LSEL.pml:56, state 49, "state = 0"
        LSEL.pml:54, state 54, "printf('Estado : Warning \n')"
        LSEL.pml:61, state 58, "-end-"
        (11 of 58 states)
unreached in proctype entorno
        (0 of 10 states)
unreached in claim p_error1
        _spin_nvr.tmp:9, state 12, "(range)"
        _spin_nvr.tmp:13, state 19, "((!((alert==1))&&range))"
        _spin_nvr.tmp:13, state 19, "(range)"
        _spin_nvr.tmp:16, state 22, "-end-"
        (3 of 22 states)

pan: elapsed time 0.127 seconds
pan: rate 606.29921 states/second
```

Al momento de realizar la verificación, se comprueba que no contiene errores, pero si observamos, existen 11 estados que no se alcanzan debido a que las condiciones que se asignó para la comprobación nunca van a llegar a determinados estados.

Decisiones de implementación

Para la implementación del sistema se optó por emplear una STM32F407VGT6 y una Raspberry Pi, ya que son componentes con los que se ha venido trabajando en anteriores asignaturas, además dispone de módulos con librerías incorporadas lo que facilita su implementación,

también cuenta con una amplia información que es accesible en internet y por su facilidad para recoger y almacenar datos de sensores en la nube y desarrollar aplicaciones IoT.

Justificación de la validez de la verificación

Como se demostró que el comportamiento del sistema está contenido en el lenguaje de Promela es decir el modelo realizado en Spin, va a cumplir la especificación.

Justificación del cumplimiento de plazos en el caso peor

	C(ms)	T(ms)	D(ms)	P
Medida de datos	138	200	300	1
Envío de datos	85	100	100	2

Los tiempos de ejecución se han obtenido midiendo la diferencia del tiempo que tardan las máquinas de estados entre los estados de IDLE o Sleep. La máquina de estados de Envío de datos tiene mayor prioridad (2) que la de Medida de datos.

Siguiendo la formula teórica obtenemos que el hiperperiodo es:

T (mcm: 100, 300) = 300ms

Debemos analizar los siguientes intervalos:

[0,100]

[0,300]

$$G\,[0{,}100] = \frac{85}{100} = 0{,}85$$

$$G\,[0{,}300] = \frac{85 + 138}{300} = 0{,}74$$

G [0,100] Max = 0.85

Por tanto, nos queda el siguiente intervalo:

[100,300]

$$G\,[100{,}300] = \frac{138}{300 - 100} = 0.69$$

f=0.85 F_{max} f = 0.69 F_{max}

0 100 300

Optimización

Partiendo de los datos obtenidos al realizar la planificación YDS, obtenemos que la intensidad en ese intervalo es 0,85 para la ejecución de la primera tarea y 0.69 para la ejecución de la segunda tarea. Partiendo de la frecuencia Máxima de la STM32F407VGT6 es de 168MHz, con esta intensidad se podría ejecutar el sistema a 142.8 MHz para el primer intervalo asignado para el primer estado y 116 MHz para el segundo estado. Se deberá escoger la frecuencia inmediata disponible.