

UNIVERSIDAD AUTONOMA DE GUADALAJARA

ETPS

Avances de Proyecto

Issac Jimenez, Juan Mejía, Paul Molina

3/25/2017

1. Nombre

ETPS

2. Descripción General del Sistema

ETPS es un sistema capaz de brindar electroterapia gracias a un circuito de potencia formado por un específico hardware, el cual tiene de salida unos electrodos para colocar al paciente, a su vez, puede estar monitoreando la presión del cuerpo por medio del dispositivo MAX30100 y estarla desplegando con un grado de fiabilidad muy alto.

En la parte de la interacción HMI, el sistema cuenta con un LCD de 16x2 caracteres con el cual informara al usuario (dependiendo del tipo de aplicación que este seleccionada) datos de un correcto funcionamiento.

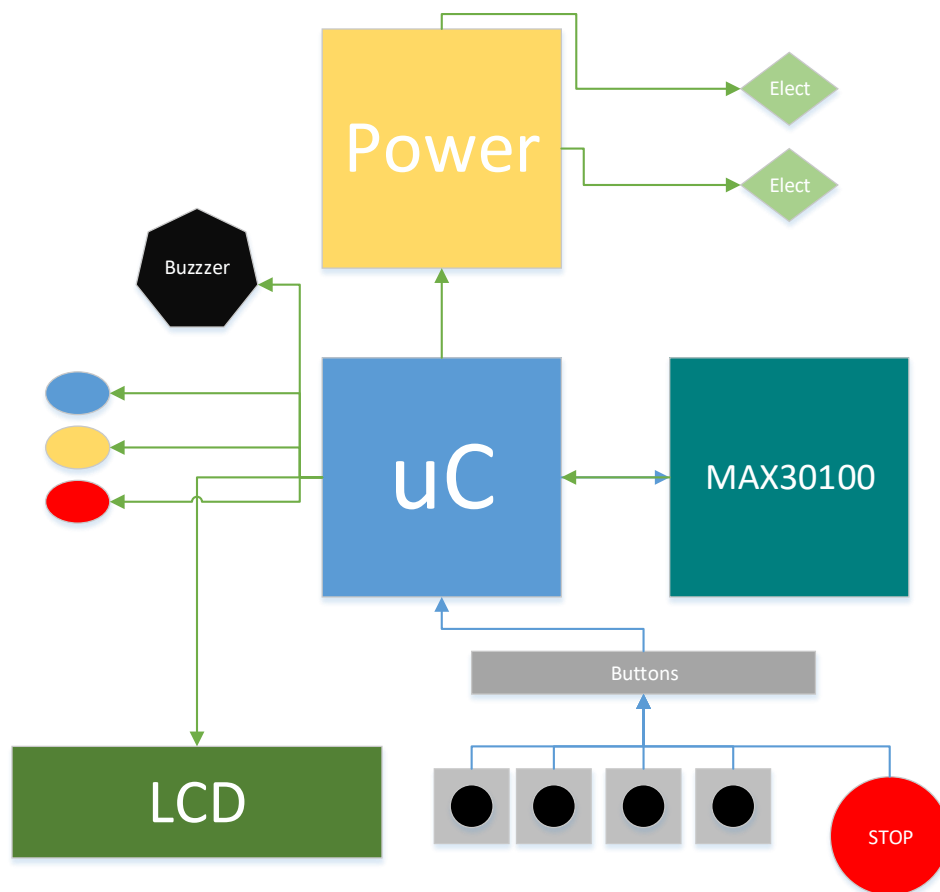


Figura 1. Diagrama General.

3. Requisitos Funcionales

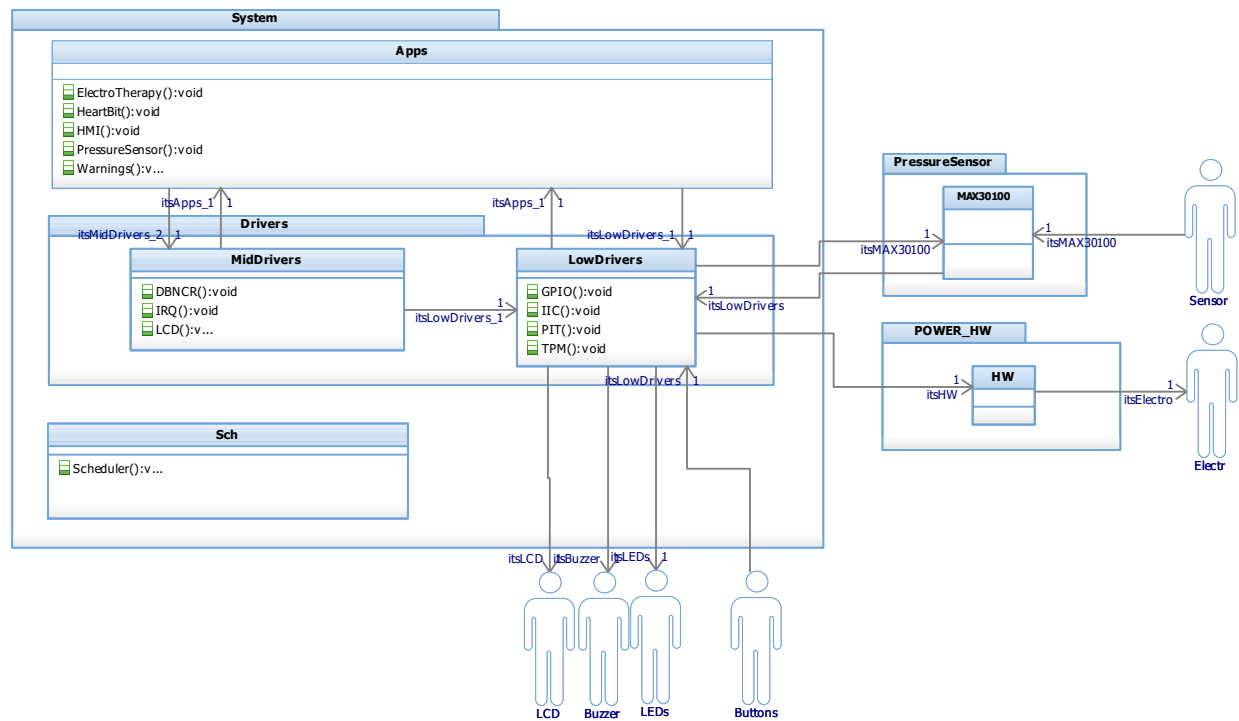


Figura 2. Diagrama de componentes y operaciones.

4. Requisitos no Funcionales

Tiempo

- El sistema de monitoreo deberá cumplir con período máximo de 10ms
- La información en el LCD se actualizará con período máximo de 100ms
- El sistema heart beat deberá cumplir con un período máximo de 500ms, y finalizara su ejecución cuando el sistema se encuentre apagado.

Fiabilidad

- El sistema deberá brindar electro estimulación, sin importar que el sistema se encuentre monitoreando la presión en ese momento.
- El sistema deberá medir la presión del paciente de una manera no invasiva.
- Las funciones de electro estimulación y monitoreo de presión son independientes y no deberán provocar interferencia, conflicto o alterar su desempeño.

Tolerancia a fallos

- En caso de emergencia el sistema tendrá un paro de emergencia, el cual detendrá las funciones de electro estimulación.
- En caso de que falle el sistema de monitoreo de presión, el sistema de electro estimulación seguirá operando.

Restricciones

- El sistema deberá ser capaz de mostrar solamente un menú a la vez
- Las funciones de monitoreo de presión y electro estimulación podrán habilitarse si el switch de encendido se encuentra en la posición de encendido.
-
- El sistema deberá apagarse por completo si el switch de encendido se encuentra en la posición de apagado.
-
- El sistema no tendrá una precisión similar a la de un dispositivo clínico.
-
- La función de electro estimulación solo se podrán configurar, posicionándose en el menú de electro estimulación.

5. Método de Diseño

Se utilizó la metodología HRT – HOOD, la cual facilita el diseño de la arquitectura lógica, la cual presentamos a continuación:

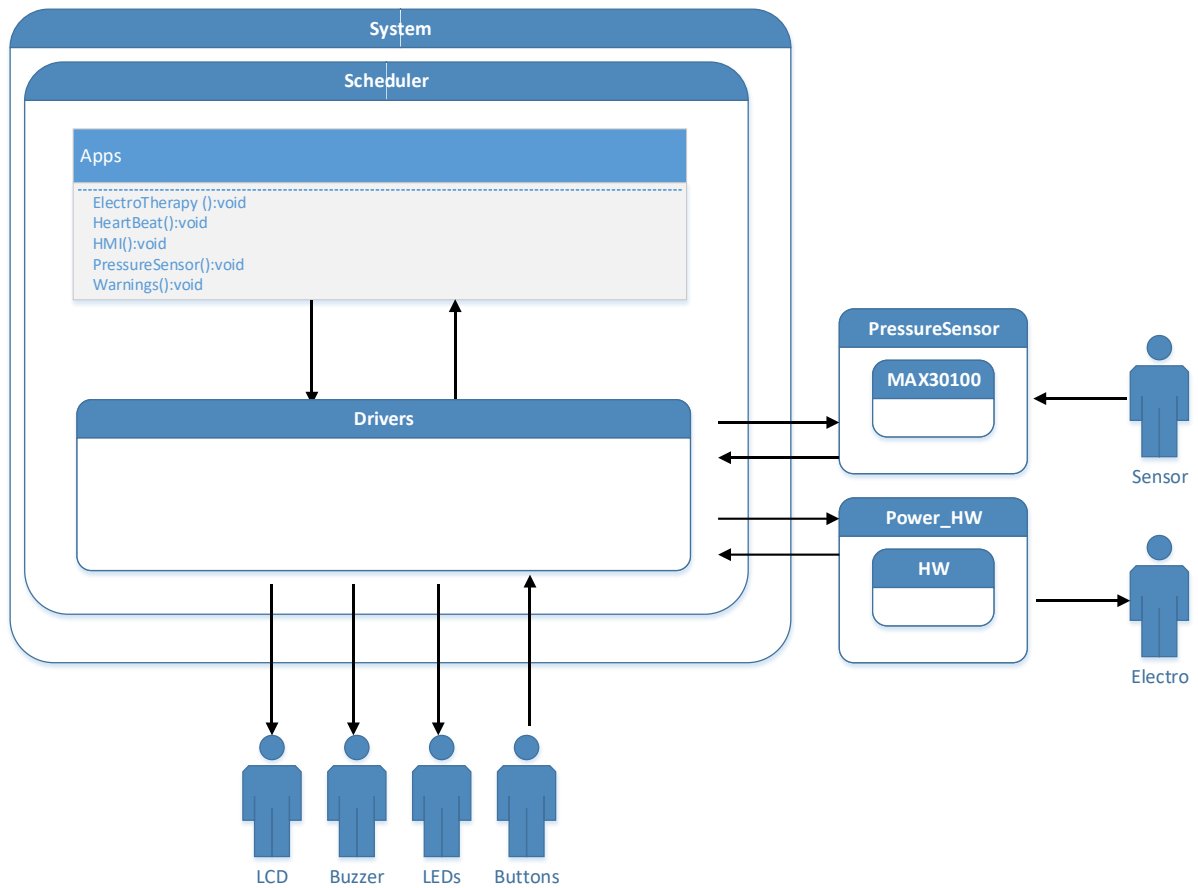


Figura 3. Arquitectura lógica del sistema.

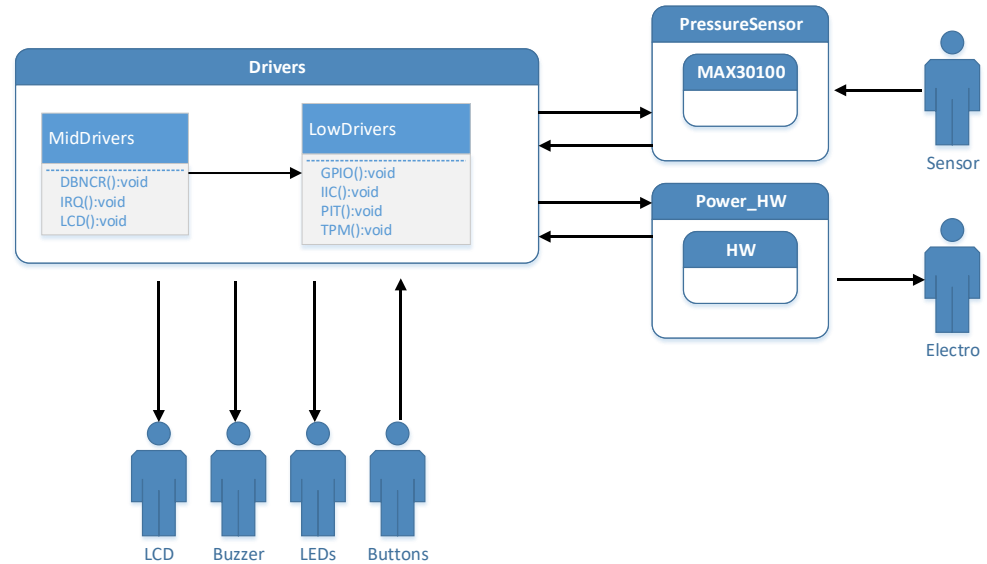


Figura 4. Arquitectura lógica del subsistema Drivers.

6. Definición de HW y SW

6.1. Definición de HW

ETPS cuenta con un Hardware tanto adquirido como hecho por nosotros a partir de simple componentes:

- Microcontrolador Kinetis KL25Z128 (NXP)
- Sensor MAX30100 para obtener la medición del pulso cardiaco
- LCD 16x2
- 5 Push Buttons
- Buzzer
- Circuito de potencia diseñado (Transistores, Diodos, Resistencias, Transformador)
- Electrodo
- LEDs (Rojo, Azul, Amarillo)

6.2. Definición de SW

La definición del software está completamente basada en los requisitos, haciendo a ETPS un sistema embebido de calidad en su respuesta y cumplimiento de tareas. Se hicieron capas 4 capas importantes en la definición del software:

- App
- MidDrivers
- LowDrivers
- System

Se realizó un análisis específico para cada necesidad del sistema y se fue agrupando por funcionalidad y características, al tener todos los componentes de software que cumplen con los requisitos, se agruparon en estas 4 capas que a su vez, cada una tiene distintos componentes

- App: (Electroterapia, HMI, Presión, Alertas, HB)
- MidDrivers: (DBNCR, IRQ, LCD)
- LowDrivers: (GPIO, IIC, PIT, TPM)
- System(Scheduler, SystemInit)

7. Análisis de Planificación

Las tareas se listan a continuación:

- Electrotherapy, (50 ms)
- Warnings(10 ms)
- HMI(100 ms)
- PressureSensor(10 ms)
- LCDDriver(5 ms)
- HeartBit(500 ms)
- Debouncer(20 ms)

No se utilizó un sistema operativo, por lo cual se diseñó un método para que las tareas se compartan el uso del procesador, la técnica utilizada es con una interrupción que se ejecuta cada 5 ms. Cada tarea tiene un contador el cual se va comparando (contra un tiempo target) y aumentándose (en el caso de no llegar al target) dentro de la interrupción, una vez que se alcanza el tiempo target se levanta una bandera, la cual da el permiso de que una tarea puede ser ejecutada.

```

14  typedef struct
15  {
16      void (*TaskFunction) (void);
17      uint16 u16CurrentTaskTime;
18      uint16 u16TargetTaskTime;
19      uint8  u8FlagToExectue;
20  }SCH_tstTaskStruct;
21
22  static SCH_tstTaskStruct _astTasks[ SCH_nTASKS] = SHC_nTASK_DEFINITION;
23
24  void SCH_vMonitor ( void )
25  {
26      uint16 u16TaskIndex = 0;
27
28      while(u16TaskIndex != SCH_nTASKS)
29      {
30          if( _astTasks[u16TaskIndex].u8FlagToExectue )
31          {
32              _astTasks[u16TaskIndex].TaskFunction();
33              _astTasks[u16TaskIndex].u8FlagToExectue = 0;
34          }
35          u16TaskIndex++;
36      }
37  }
38

```



```

18  #define SCH_nTASKS 3
19
20
21  /*!Base Time 5 miliseconds*/
22  /*!Here define your task as:
23   *
24   *Task_Function
25   *Initial Time
26   *Target Time
27   *Execute Flag Status
28   */
29
30  #define SHC_nTASK_DEFINITION \
31  { \
32      { \
33          HB_vTask, \
34          0, \
35          100, \
36          0 \
37      }, \
38      { \
39          DBNCR_vMonitor, \
40          0, \
41          4, \
42          0 \
43      }, \
44      { \
45          HB_vTaskAnotherLed, \
46          0, \
47          50, \
48          0 \
49      } \
50  }

```

```

39  void PIT_IRQHandler(void)
40  {
41      uint16 u16TaskIndex = 0;
42
43      if (PIT_TFLG0 & PIT_TFLG_TIF_MASK)
44      {
45          PIT_TFLG0 |= PIT_TFLG_TIF_MASK;
46
47          while(u16TaskIndex != SCH_nTASKS)
48          {
49              if( _astTasks[u16TaskIndex].u16CurrentTaskTime == (_astTasks[u16TaskIndex].u16TargetTaskTime - 1) )
50              {
51                  _astTasks[u16TaskIndex].u8FlagToExectue = 1;
52                  _astTasks[u16TaskIndex].u16CurrentTaskTime = 0;
53              }
54              else
55              {
56                  _astTasks[u16TaskIndex].u16CurrentTaskTime++;
57              }
58
59              u16TaskIndex++;
60          }
61      }
62  }

```

| Tarea | Tiempo de Computo | Periodo | U |
|----------------|----------------------|---------|------|
| LCD Driver | 1 ms | 5 ms | .2 |
| Warnings | 1 ms | 10 ms | .1 |
| Pressure | 1 ms | 10 ms | .1 |
| Debouncer | 1 ms | 20 ms | .05 |
| ElectroTherapy | 1 ms | 50 ms | .02 |
| HMI | 1 ms | 100 ms | .01 |
| HeartBlt | 1 ms | 500 ms | .002 |
| Total U | | | .482 |

[illegible]