Apéndice A

Programas y Diagramas de Bloques para los Microprocesadores

Microprocesador PIC16F84A (Transmisor):

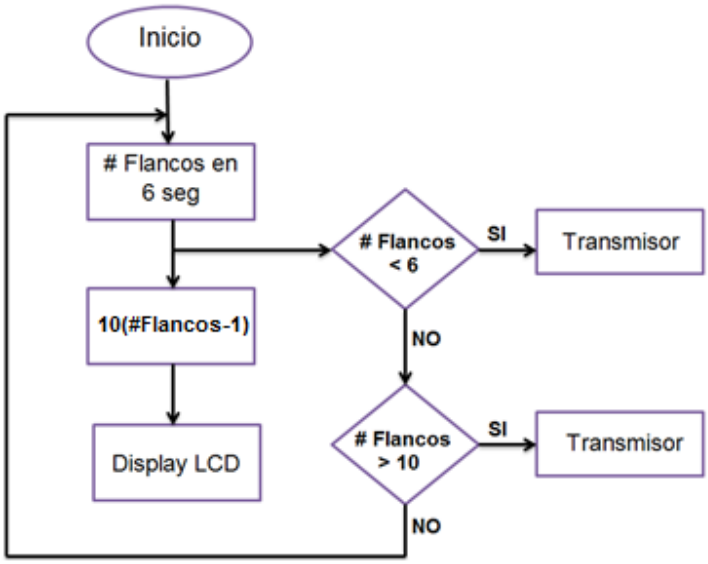


Figura A.1 Diagrama de bloques del PIC16F84A en transmisor

```
;*********************************Transmisor.asm*******************************
;
; ZONA DE DATOS:
;
            LIST      P=16F84A
            INCLUDE      <P16F84A.INC>
            __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON &_XT_OSC;   CONFIGURACION PARA
EL GRABADOR

            CBLOCK     0X0C
            NO_FLANCOS
            GuardaNumero
            ENDC
;
; ZONA DE CODIGO:
;
        ORG          0
;
INICIO      call              LCD_Inicializa;
            clrf              TRISB
            bsf               STATUS,RP0          ; acceso al banco 1
            clrf              TRISB               ; se limpia el puerto B
            movlw             b'00111000'         ; se carga el registro W
            movwf             OPTION_REG          ; configuración como contador, flancos descendente
            bcf               STATUS,RP0          ; se cierra, acceso al banco 0
```

```
PRINCIPAL
        clrf        TMR0            ; inicialización del Timmer
        call        Retardo_5s      ; 6 segundos para contar el número de flancos
        call        Retardo_1s
        movf        TMR0, W         ; TMR0→W
        movwf       NO_FLANCOS      ; W → NO_FLANCOS
        call        ciclosRR        ; Llamado a subrutina ciclos RR
        call        Multiplicacion  ; Llamado a subrutina Multiplicación
        call        VisualizaNumero ; Llamado a subrutina VisualizaNumero
        movlw       Mensaje0
        call        LCD_Mensaje     ; Llamado a subrutina que muestra el # de LPM
        call        Retardo_1s;     ; espera de 1 segundo
        call        LCD_Borra       ; se borra la pantalla para mostrar un nuevo valor

        ; Comparaciones:

        MOVLW       D'6'            ; Valor a comparar
        SUBWF       NO_FLANCOS, W   ; NO_FLANCOS – 6 → W
        BTFSS       STATUS, C       ; compara el resultado de la diferencia
        CALL        alerta1         ; llamado a la subrutina alerta 1

        movlw       D'10'           ; comparación para lpm > 100
        subwf       NO_FLANCOS, W   ; NO_FLANCOS – 12 → W
        btfsc       STATUS, C       ;
        call        alerta2         ; llamado a subrutina alerta 2
        goto        PRINCIPAL       ; regresa a Principal

; Subrutina de ciclosRR
        movlw   D'1'                ; se carga a W con 1
        subwf   NO_FLANCOS          ; NO_FLANCOS +1 →NO_FLANCOS
        btfss   STATUS,C            ; compara el resultado de la diferencia
        addwf   NO_FLANCOS          ; NO_FLANCOS +1 → NO_FLANCOS
        return                      ;retorna el valor del No. de ciclos RR

; Subrutina De Multiplicación:
Multiplicacion
        movf        NO_FLANCOS,W    ;  NO_FLANCOS → W
        addwf       NO_FLANCOS,0    ; se suma el valor de número de flancos
        addwf       NO_FLANCOS,0    ; con el valor de W
        addwf       NO_FLANCOS,0    ; repitiendo el proceso unas 10 veces
        addwf       NO_FLANCOS,0    ;
        addwf       NO_FLANCOS,0    ;
        addwf       NO_FLANCOS,0    ;
        addwf       NO_FLANCOS,0    ;
        addwf       NO_FLANCOS,0    ;
        addwf       NO_FLANCOS,0    ;
        return                      ; retorna el valor de la multiplicación
```

```
; Subrutina para alerta:
alerta1
        bsf         PORTB,0             ; enciende al pin 0 del puerto B
        call        Retardo_1s          ; lo deja prendido un segundo
        bcf         PORTB,0             ; apaga al pin 0 del puerto B
        return                          ; regresa al cuerpo principal del programa

; Subrutina para alerta2:
alerta2
        bsf         PORTB,1             ; enciende al pin 1 del puerto B
        call        Retardo_1s          ; lo deja prendido un segundo
        bcf         PORTB,1             ; apaga al pin 1 del puerto B
        return                          ; regresa al cuerpo principal del programa

; Subrutina para visualizar el número de latidos:
VisualizaNumero
        Movwf       GuardaNumero        ; Reserva el número.
        call        BIN_a_BCD           ; Pasa el número a BCD.
        movf        BCD_Centenas,W      ; Primero las centenas.
        btfss       STATUS,Z            ; Si son cero no visualiza las centenas.
        goto        VisualizaCentenas
        movf        GuardaNumero,W      ; Vuelve a recuperar este valor.
        call        BIN_a_BCD           ; Lo pasa a BCD.
        call        LCD_Byte            ; Visualiza las decenas y unidades.
        goto        FinVisualizaNumero
VisualizaCentenas
        call        LCD_Nibble          ; Visualiza las centenas.
        movf        GuardaNumero,W      ; Vuelve a recuperar este valor.
        call        BIN_a_BCD           ; Lo pasa a BCD.
        call        LCD_ByteCompleto    ; Visualiza las decenas aunque sea cero.
FinVisualizaNumero
        return

; Subrutina para mensaje en pantalla:
 Mensajes
                addwf   PCL,F;
      Mensaje0
                DT"  Lat/Min",0x0       ; mensaje para mostrar las unidades en pantalla
                                        ; LCD

        INCLUDE<LCD_4BIT.INC>
        INCLUDE<BIN_BCD.INC>
        INCLUDE<LCD_MENS.INC>
        INCLUDE<RETARDOS.INC>
        END
```

Microprocesador PIC16F84A (Receptor):

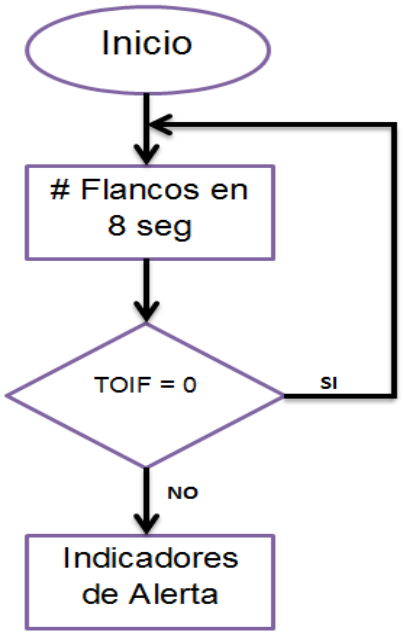


Figura A.2 Diagrama de bloques del PIC16F84A en receptor

```
;*********************************RECEPTOR.asm***********************************
;
;
; ZONA DE DATOS:
;
              LIST      P=16F84A
              INCLUDE      <P16F84A.INC>
              __CONFIG _CP_OFF & _WDT_OFF & _PWRTE_ON &_XT_OSC ; CONFIGURACION PARA
;EL GRABADOR

              CBLOCK     0X0C
              NO_FLANCOS
              ENDC
; ZONA DE CODIGO:

              ORG          0

INICIO        call              LCD_Inicializa
              clrf              TRISB
              bsf               STATUS,RP0          ; acceso al banco 1
              clrf              TRISB               ; se limpia el puerto B
              movlw             b'00111000'         ; se carga el registro W
              movwf             OPTION_REG          ; configuración como contador
              bcf               STATUS,RP0          ; se cierra, acceso al banco 0
```

```
PRINCIPAL

        clrf    TMR0            ; inicialización del Timmer
        call    Retardo_5s      ; espera de 8 segundos
        call    Retardo_2s;
        call    Retardo_1s
        btfss   INTCON,2        ; Revisión de bit para control de desborde en TMR0
        goto    PRINCIPAL       ; sino se ha desbordado, regresa a PRINCIPAL
        bcf     INTCON,2        ; Vuelve a poner en cero al bit INTCON
        CALL    ALERTA          ; llamado a la subrutina alerta
        GOTO    PRINCIPAL


; SUBRUTINA DE ALERTA:

ALERTA
        movlw   Mensaje0        ; etiqueta para el mensaje a mostrar en pantalla LCD
        call    LCD_Mensaje     ; subrutina que muestra el mensaje en la pantalla LCD

        BSF     PORTB,0
        BSF     PORTB,3
        CALL    Retardo_500ms       ; SE PRENDEN DURANTE 1 SEG.

        BCF     PORTB,0
        BCF     PORTB,3
        CALL    Retardo_500ms       ; SE APAGA DURANTE 1 SEG(1)

        BSF     PORTB,0
        BSF     PORTB,3
        CALL    Retardo_500ms       ; SE PRENDEN DURANTE 1 SEG.

        BCF     PORTB,0
        BCF     PORTB,3
        CALL    Retardo_500ms       ; SE APAGA DURANTE 1 SEG(2)

        BSF     PORTB,0
        BSF     PORTB,3
        CALL    Retardo_500ms       ; SE PRENDEN DURANTE 1 SEG.

        BCF     PORTB,0
        BCF     PORTB,3
        CALL    Retardo_500ms       ; SE APAGAN DURANTE 1 SEG(3)

        BSF     PORTB,0
        BSF     PORTB,3
        CALL    Retardo_500ms     ; SE PRENDEN DURANTE 1 SEG.
```

```
            BCF             PORTB,0
            BCF             PORTB,3
            CALL            Retardo_500ms       ; SE APAGAN DURANTE 1 SEG(4)

            BSF             PORTB,0
            BSF             PORTB,3
            CALL            Retardo_500ms       ; SE PRENDEN DURANTE 1 SEG.


            BCF             PORTB,0
            BCF             PORTB,3
            CALL            Retardo_500ms       ; SE APAGAN DURANTE 1 SEG(5)

            BSF             PORTB,0
            BSF             PORTB,3
            CALL            Retardo_500ms       ; SE PRENDEN DURANTE 1 SEG.

            BCF             PORTB,0
            BCF             PORTB,3
            CALL            Retardo_500ms       ; SE APAGAN DURANTE 1 SEG(6)

            CALL            LCD_Borra
            RETURN

; Mensaje de Alerta

 Mensajes
                addwf    PCL,F
        Mensaje0
                DT" ALERTA   ALERTA",0x0          ; mensaje que se verá en el display LCD

        INCLUDE<LCD_4BIT.INC>
        INCLUDE<LCD_MENS.INC>
        INCLUDE<RETARDOS.INC>
        END
```

Microprocesador PICAXE 14-M (Adquisidor de datos):

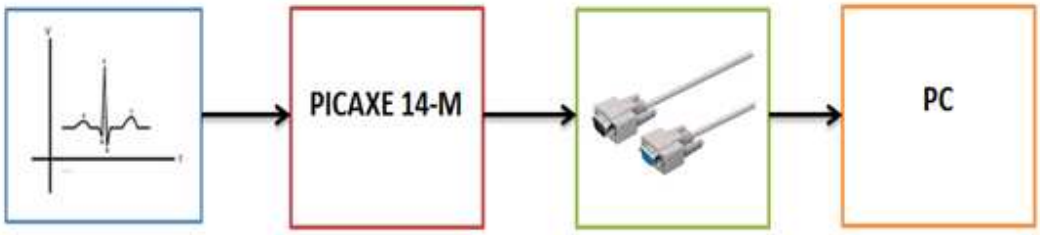
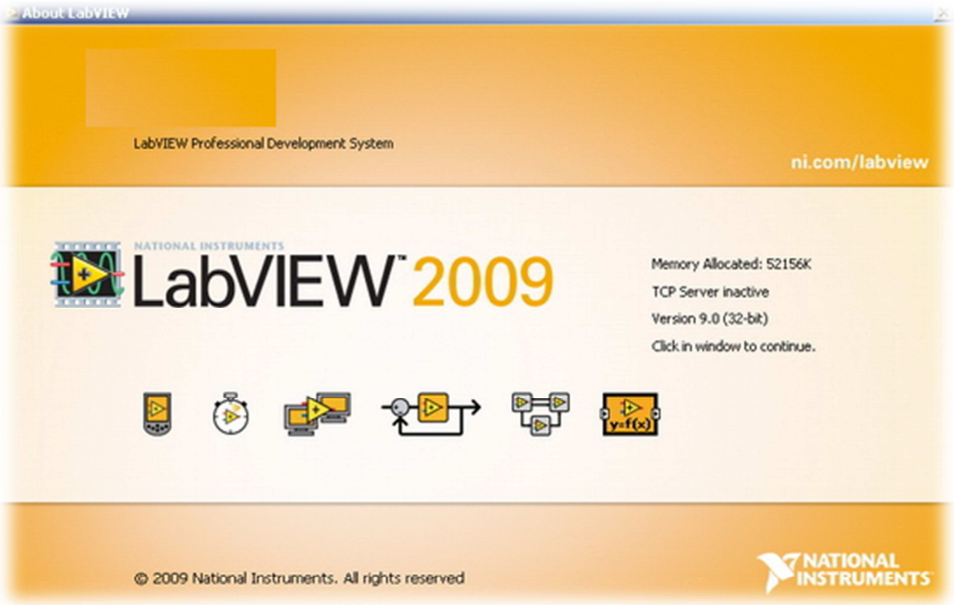


Figura A.3 Diagrama de bloques del adquisidor

```
main:   readadc 0,b0          ; leer señal del canal 0 en la variable b0
        serout 5,N4800,(b0)   ; transmite datos via serial por el pin de salida
                              ; a una velocidad  de 4800 baudios
        goto main             ; salta a main
```

Apéndice B

Programación en LabVIEW 9.0



Programación en LABVIEW:

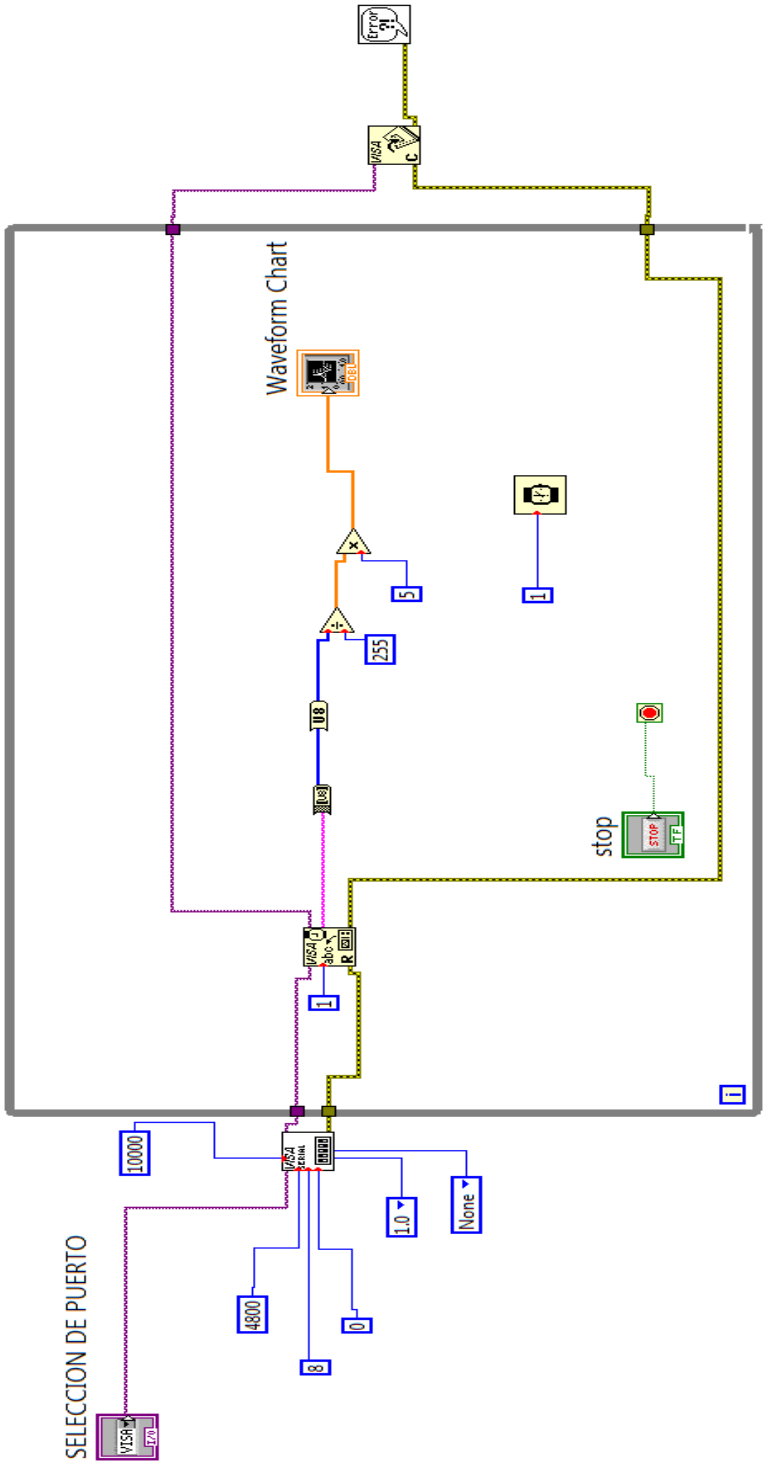


Figura B.1 Programación del monitor en Labview 9.0

La programación de la interfaz visual (monitor) se basa en los siguientes elementos [20]:

- **Visa Resource Name:** 

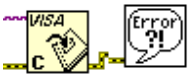
nos permite escoger el puerto COM sobre el cual conectaremos al dispositivo.

- **Visa Serial:** 

configura la comunicación serial en Labview, se introducen aquí los siguientes datos: 8 bits, sin bit de paridad, 1 bit de parada, 4800 baudios.

- **Visa Read:** 

lee la información proveniente del puerto seleccionado.

- **Visa C:** 

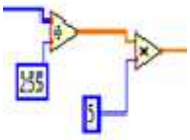
si existió algún error en la lectura de Visa Read, cerrara el evento de lectura y mostrara un mensaje de error.

- **Convertidor de Cadena a Byte:** 

a la salida de Visa Read, la información está en forma de cadena. Por lo tanto se usa un convertidor de cadena a Byte sin signo.

- **Convertidor de Byte a Entero:** 

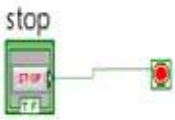
en forma de Byte no se puede mostrar la información en la gráfica, por lo que se emplea un convertidor de Byte a entero, cuyo valor ahora si es utilizable en la gráfica.

- **Escala de la gráfica:** 

la escala está dada por un división entre 255 y una multiplicación por 5. Se divide entre 255, porque es el valor máximo digital que asigna el convertidor al máximo valor analógico de la señal introducida. Se multiplica por 5 para dejar a la señal en una amplitud de 5 volts como máximo.

- **Gráfica:** 

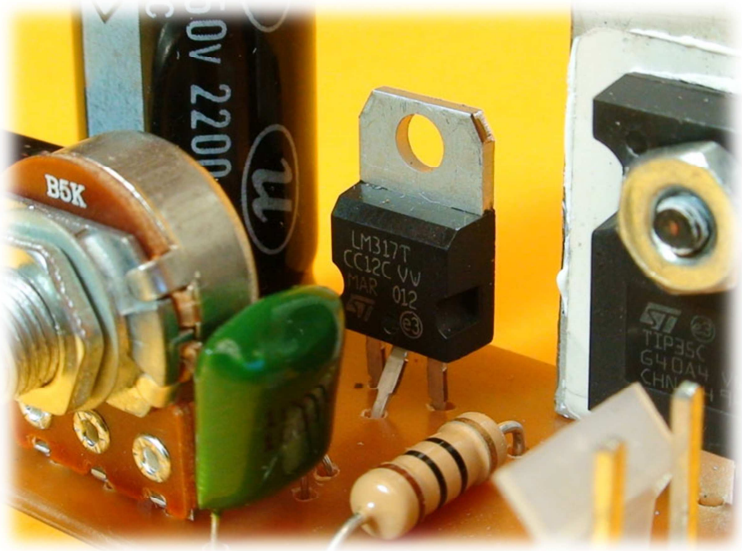
este elemento grafica los valores que provienen del convertidor de Byte a Entero sin signo.

- **Stop:** 

detiene la adquisición de datos.

Apéndice C

Fuentes de Alimentación



La alimentación del electrocardiógrafo consta de dos fuentes de voltaje: una hecha a base de baterías y la segunda es una fuente de voltaje rectificada.se dan estas dos opciones para darle alternativas al usuario, ya que en el caso más pesimista puede no haber conexión alguna para alimentar a la fuente rectificada, quedando como única opción la fuente de baterías.

Y en contra parte, en caso de no tener baterías o ahorrar en baterías, está la opción de poder emplear la fuente rectificada. El fin es que donde sea que se mueva el electrocardiógrafo pueda operar. A continuación se desglosa cada una de las fuentes.

Fuente rectificada: aquí se convierte la señal alterna del suministro eléctrico en una señal de directa, mediante procesos de rectificación y filtrados [21], para tener voltajes de salida de +/- 15V y 5V.

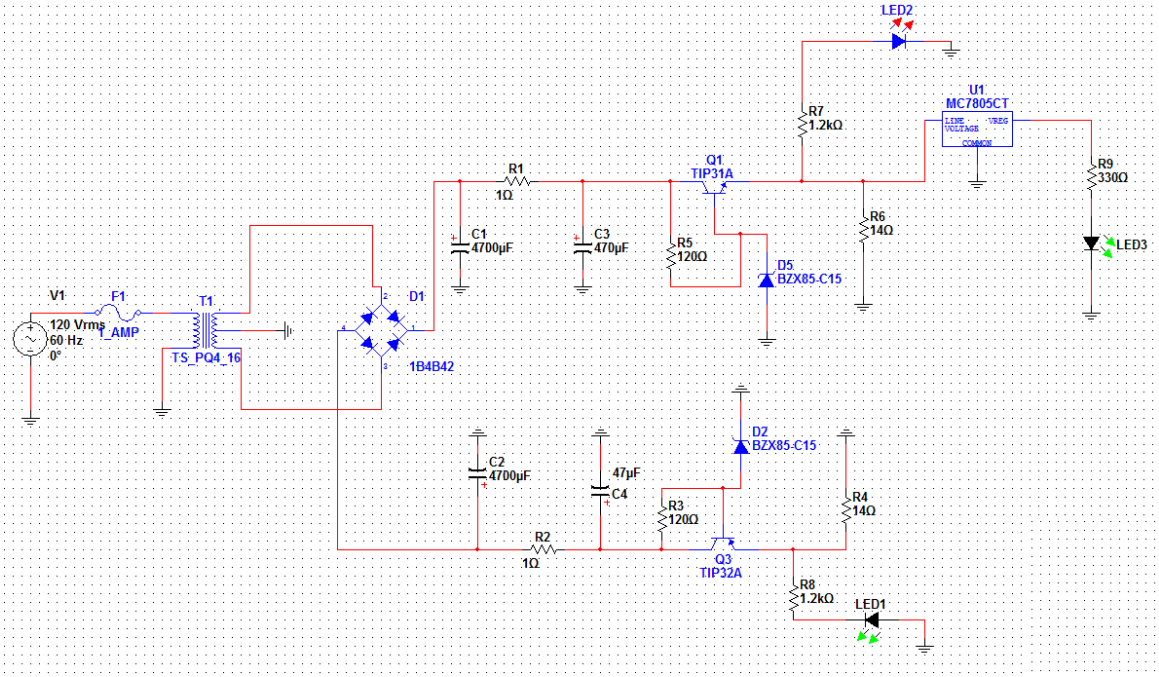


Figura C.1 Fuente de voltaje rectificada

Medición de los voltajes a la salida:

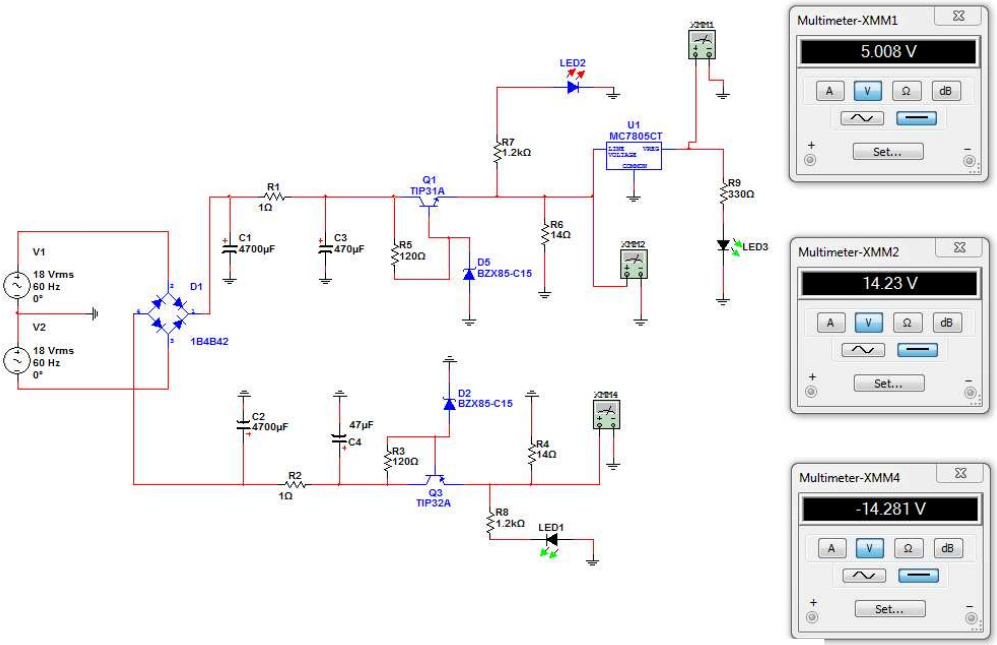


Figura C.2 Medición de voltaje en las salidas

Medición de corriente en la parte positiva y negativa de la fuente:

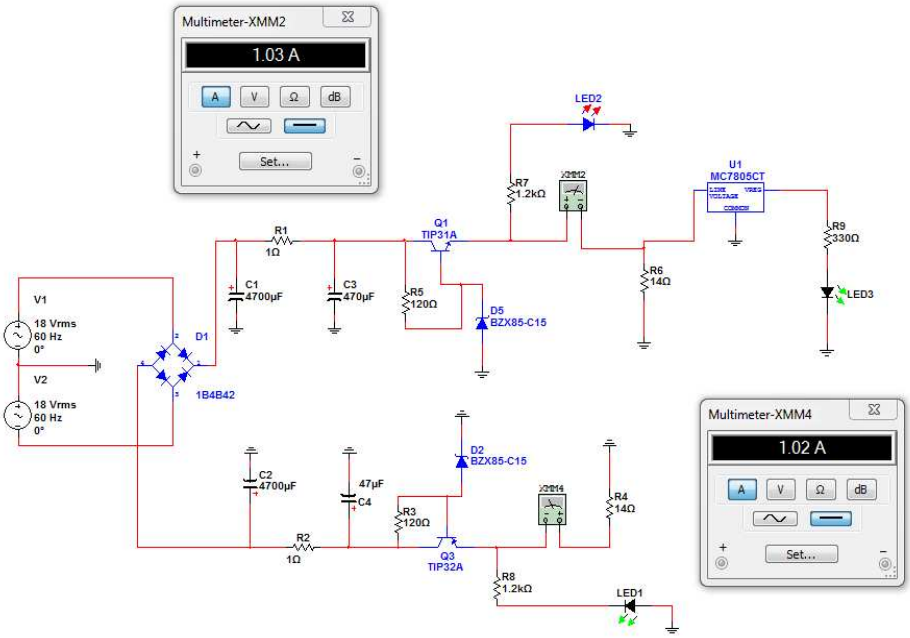


Figura C.3 Medición de corriente en las salidas

Las características de esta fuente son: Vo=+/-14.3V y 5V, Io=1A, transformador reductor de 120V a 32V con tap central, puente rectificador de 200V a 1A, capacitores electrolíticos a 50V, diodo zener 15V a 1 watt, resistencia de carga a 1 watt.

Fuente en base a baterías: se usan dos pilas comunes de 9V y un regulador de 5V, teniendo como voltajes de alimentación +/- 9V y +5V, como se muestra en la figura C.2.

Se agrupa la fuente de baterías y la fuente rectificada, se coloca un switch de un polo y un tiro, y otro switch de dos polos y dos tiros, esto permitirá habilitar las fuentes. Por ejemplo en la Figura C.4, si se deja abierto S2B y S1B se cierra, la fuente rectificada alimentaria al electrocardiógrafo y transmisor; por el lado contrario si se cierra S2B y se deja abierto S1B, las baterías alimentarían al dispositivo.

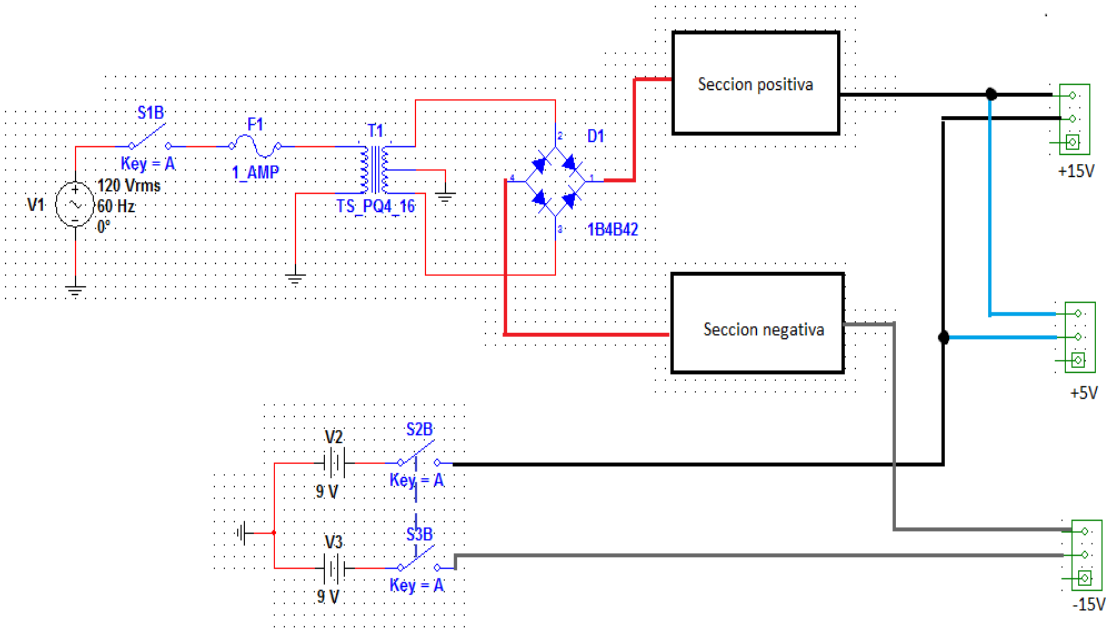


Figura C.4 Fuente rectificada y de baterías

Fuente de alimentación para el receptor y 2do. PIC16F84A

Es una fuente muy sencilla para el circuito receptor y segundo PIC16F84A (cerebro del circuito indicador de alerta), debido a que tiene pocos elementos (display LCD, PIC16F84A, diodo IR). La fuente constara de una batería de 12V a 1A junto con un regulador de 5 volts y como protección un diodo, para evitar la inversión de polaridades.

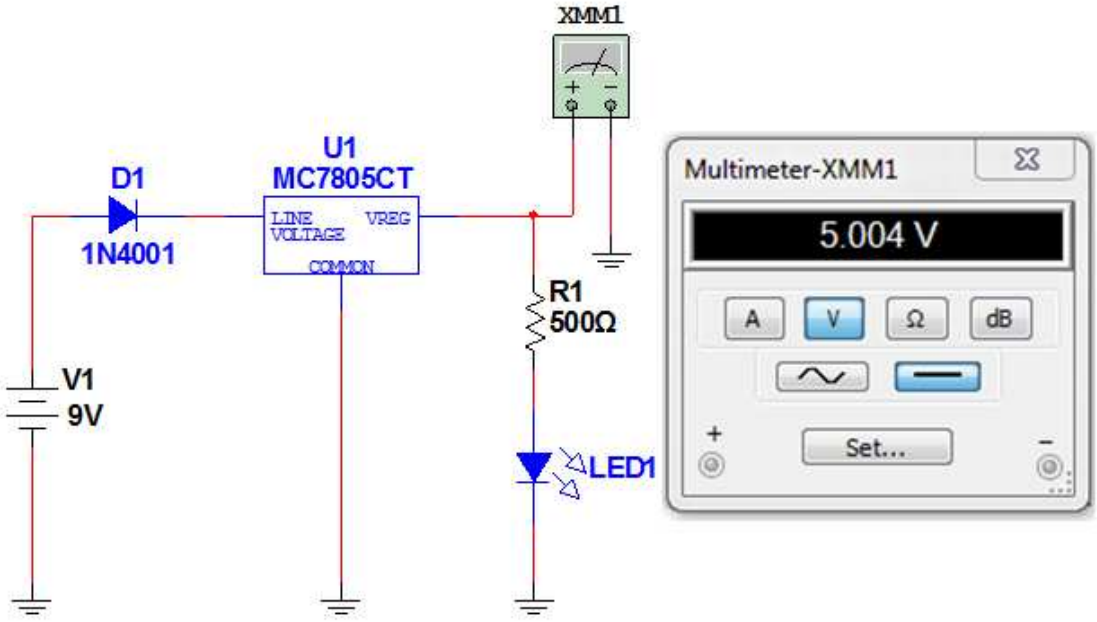


Figura C.5 Fuente de voltaje para receptor y 2do PIC16F84A