

DSE G4

Control de un brazo robótico mediante la placa PIC24

Autor: Gorka Dabó

Fecha: 08/05/2024

Instrucciones de uso

Conexiones

- Conectar el cable azul "Rx" al pin RB8.
- Conectar el cable amarillo "Ry" al pin RB9.
- Conectar el cable blanco "Palanka" al pin R10.
- Conectar el cable marrón "urrekoia" al pin RB2.
- Conectar el cable rojo a 3.3V.
- Conectar los 2 cables negros a GND.
- Conectar el cable S1 al pin RD0.
- Conectar el cable S2 al pin RD1.
- Conectar el cable S3 al pin RD2.
- Conectar el cable S4 al pin RD3.
- Conectar el cable S5 al pin RD9.

Funcionamiento

El sistema integra 3 botones, un potenciómetro, una pantalla LCD, un termómetro, un joystick, un brazo robótico y el teclado. A continuación se describe la función de cada componente:

Botones

- **S3:** Realiza scroll sobre el contenido de la pantalla LCD, mostrando la siguiente línea.
- **S4:** Coloca el brazo en una posición segura.
- **S6:** Alterna el control del brazo robótico entre el joystick y la UART.

Otros componentes

- **Potenciómetro:** Controla el servomotor 4, que mueve la pinza hacia arriba o abajo.
- **Pantalla LCD:** Muestra información sobre el estado del brazo robótico, joystick, cronómetro, potenciómetro y termómetro.
- **Termómetro:** Mide la temperatura ambiente.
- **Joystick:** Controla los ejes X, Y con el propio joystick y el eje Z con una palanca lateral. El gatillo frontal cierra la pinza del brazo.
- **Brazo robótico:** Sistema de 5 servomotores: 4 controlados por joystick y 1 por potenciómetro (o todos por teclado si se habilita UART).

Control por teclado

El teclado permite controlar el brazo robótico y el cronómetro mediante las siguientes teclas (mayúsculas o minúsculas):

Tecla	Función
P	Para el cronómetro
C	Continúa el cronómetro
T	Reinicia el cronómetro
D / A	Mueve el brazo hacia la derecha / izquierda
W / S	Mueve el brazo hacia arriba / abajo
T / G	Mueve el brazo en el eje Z hacia delante / atrás
U / J	Mueve la pinza hacia arriba / abajo
K / H	Cierra / abre la pinza

Características

1. Dispositivos periféricos y módulos utilizados

Dispositivo	Identificador
LEDs	D7
Pulsadores	S3, S4, S6
Temporizadores	T2, T4, T8
Pantalla LCD	-
Convertidor Analógico/Digital	ADC1
UART	UART2
Módulo CN	-

2. Tipo de sincronización con los periféricos

- Los pulsadores S4, S3 y S6 están sincronizados por interrupción.
- Los temporizadores T2 y T4 se sincronizan por interrupción.
- El UART2 está sincronizado por interrupción.
- El módulo ADC1 se sincroniza por interrupción.
- El módulo CN se sincroniza por interrupción.

3. Funciones del programa principal

inic_oscilator

Fichero: Oscilator.c

Selección e inicialización del reloj a 80 MHz.

inic_joystick

Fichero: GPIO.c

Activa los pines necesarios para el uso del joystick: RB2, RB8, RB9 y RB10.

inic_brazo

Fichero: GPIO.c

Activa los pines necesarios para el brazo robótico: RD0, RD1, RD2, RD3 y RD9.

inic_Timer4()

Fichero: timers.c

Inicializa el Timer 4 para generar interrupciones cada 4 ms, configurando el registro de periodo, el prescale y habilitando la interrupción asociada.

Init_LCD()

Fichero: LCD.c

Inicializa la pantalla LCD con una secuencia de configuración que incluye un retraso de 15 ms para estabilizar el voltaje y ejecuta comandos de configuración inicial.

inic_leds()

Fichero: GPIO.c

Configura los 8 pines de PORTA (RA0-RA7) como salidas para LEDs y los apaga inicialmente.

inic_ADC1()

Fichero: ADC1.c

Inicializa el convertidor analógico/digital para realizar 2000 conversiones por segundo, configurando los pines RB2, RB4, RB5, RB8, RB9 y RB10 como entradas analógicas.

inic_pulsadores()

Fichero: GPIO.c

Activa las entradas de los pulsadores S3, S4, S5 y S6.

inic_crono()

Fichero: timers.c

Inicializa las variables del cronómetro: mili, deci, seg y min.

inc_UART2()

Fichero: UART2_RS232.c

Inicializa el módulo UART2 configurando la velocidad de transmisión, habilitando el módulo y las interrupciones de transmisión y recepción.

inic_CN()

Fichero: CN.c

Inicializa el módulo CN que posibilita la utilización de los pulsadores.

inic_pwm()

Fichero: GPIO.c

Inicializa los pines de salida de los servomotores con valor 0.

inic_Timer2()

Fichero: timers.c

Inicializa el timer utilizado para el control de múltiples servomotores, cuya función de interrupción se encarga de cambiar los valores PR2 y LATDbits para mover los motores correctamente.

4. Rutinas de atención a interrupciones

_ADC1Interrupt()

Fichero: ADC1.c

Recoge el valor digitalizado del potenciómetro, sensor de temperatura y joystick. En cada iteración guarda el valor del muestreo de una entrada. Tras 8 iteraciones de guardar los valores del potenciómetro, sensor de temperatura y ejes X, Y, Z del joystick, activa el flag que calculará la media. Finalmente reactiva SAMC para realizar un nuevo muestreo.

_T2Interrupt()

Fichero: timers.c

Controla el movimiento de los servomotores. Cada vez que se interrumpe, se mueve un servomotor de forma secuencial, manipulando los valores de duties[] que contienen el nuevo valor del servomotor (y el que se pondrá en PR2).

_T4Interrupt()

Fichero: timers.c

Suma 4 a la variable 'mili' para el funcionamiento del cronómetro.

_U2RXInterrupt()

Fichero: UART2_RS232.c

Observa qué tecla ha sido presionada y según ella ejecuta una función u otra. Entre estas funciones están las que manipulan el cronómetro (reiniciando, pausando, deteniendo) y las que manipulan los servomotores.

_U2TXInterrupt()

Fichero: UART2_RS232.c

Escribe en la pantalla de Tera Term (u otro programa similar) todos los mensajes contenidos en la variable 'Información', incluyendo el nombre del proyecto, mensaje de ayuda, cronómetro y valores de joysticks y servomotores.

_CNInterrupt()

Fichero: CN.c

Verifica el valor de los registros de RB2, RD6, RD13 y RD7. Si se han presionado, llama a funciones que cierran la pinza, realizan scroll en la pantalla LCD, ponen el brazo en posición segura o desactivan el joystick para recibir comandos por UART.

Funcionamiento del programa completo

El programa inicia inicializando todos los flags a 0. Posteriormente, inicializa el oscilador, los pines del brazo y del joystick, los timers y la pantalla LCD. Una vez completada la inicialización, el programa entra en un bucle while(1) donde verifica constantemente qué flags se han activado.

En este punto, la placa, el teclado y el joystick están operativos. El joystick permite mover el brazo robótico en los 3 ejes, además de controlar los dos servomotores de

la pinza: uno mediante el potenciómetro y otro con el gatillo del joystick. El UART puede utilizarse únicamente para controlar el cronómetro (parar, inicializar y continuar) y mostrar información en pantalla.

El botón S4 permite mover el brazo robótico a una posición segura. Al presionar el botón S6, el brazo robótico deja de recibir movimiento del joystick y comienza a recibirllo por la UART, manteniendo el funcionamiento inicial.

Conclusiones

Este proyecto ha permitido consolidar los conocimientos adquiridos en todas las prácticas anteriores. Aunque supuso un reto volver a desarrollar todo desde cero para evitar conflictos, el resultado fue muy beneficioso al mejorar significativamente la calidad del código y el funcionamiento general del sistema.

Gracias a este enfoque metódico, se han minimizado los problemas habituales, logrando que el brazo robótico funcione sin complicaciones significativas.

En conclusión, en este proyecto se ha conseguido controlar el funcionamiento de un brazo robótico utilizando un joystick y el teclado de un ordenador. Para ello, se han controlado los pines de entrada/salida y los servomotores utilizando el módulo ADC y el Timer 2.