

Documentazione Codice MSP432: Lettura Sensori e Invio Dati via UART

Introduzione

Questo codice è stato sviluppato per il microcontrollore MSP432 e ha lo scopo di leggere i dati da un joystick (X, Y) e da un potenziometro utilizzando l'ADC (Analog-to-Digital Converter). I dati acquisiti vengono trasmessi via UART solo se si verifica una variazione rispetto al valore precedente. Inoltre, è stato implementato un sistema di notifica con LED che segnala ogni evento di input.

Struttura del Codice

1. Definizione della Struttura dei Dati

Il codice utilizza una struttura `SensorData` per contenere il tipo di azione e il valore misurato.

```
typedef struct {  
    int actionType; // 0 = Joystick X, 1 = Joystick Y, 2 = Potenziometro, 3-6 = Pulsanti  
    float value;  
} SensorData;
```

2. Configurazione della UART

La comunicazione seriale è configurata con una velocità di 115200 baud per trasmettere i dati in modo efficiente.

```
const eUSCI_UART_ConfigV1 uartConfig = { ... };
```

3. Gestione dell'ADC

L'ADC è inizializzato per acquisire i dati dal joystick (X, Y) e dal potenziometro. I valori acquisiti

vengono normalizzati e confrontati con quelli precedenti per evitare invii ridondanti.

```
if (xPosition != prevXPosition) {  
    SensorData joystickData = {0, xPosition};  
    sendUARTData(joystickData);  
    prevXPosition = xPosition;  
}
```

4. Gestione degli Interrupt

Gli interrupt vengono utilizzati per catturare i dati dal joystick e dal potenziometro e per rilevare la pressione dei pulsanti.

```
void ADC14_IRQHandler(void) { ... }  
void PORT4_IRQHandler(void) { ... }
```

5. Accensione LED su Input

Un LED è acceso quando si verifica un qualsiasi evento di input e spento dopo l'elaborazione.

```
GPIO_setOutputHighOnPin(GPIO_PORT_P1, GPIO_PIN0); // Accende LED
```

Conclusione

Questo codice garantisce un'efficiente gestione degli input del joystick e del potenziometro, trasmettendo solo le variazioni dei dati via UART e fornendo un riscontro visivo attraverso un LED. È progettato per funzionare in modalità a basso consumo con l'uso di `PCM_gotoLPM0()`.