Ultrasound Impurity Detection Device

Code Implementation, North Carolina Agricultural & Technical State Univ.

Using Bluetooth as a Controller Area Network (CAN) replacement is a bit unconventional, as CAN is a widely used communication protocol in automotive and industrial applications due to its reliability and real-time capabilities making it a suitable means of communication for our device.

The HC-05/HC-06 are standalone modules used to provide Bluetooth functionality and can be easily interfaced with the MSP430 microcontroller. Some popular Bluetooth modules include:

Below is the implementation of the Bluetooth communication between two MSP430 microcontrollers, UART communication protocol was used in the development of the device to establish secure communication of data transferring between device and application, without signal interference.

1. Configure one MSP430 microcontroller as a transmitter (sending data) and the other as a receiver (receiving data).
2. Use a Bluetooth module (such as HC-05 or HC-06) connected to each MSP430 microcontroller to establish a wireless serial connection.
3. Define a simple communication protocol to exchange data between the two devices. For example:
    - Start of Message (SOM) byte: Indicates the start of a message.
    - Message ID: Identifies the type of message being sent.
    - Data: Payload of the message.
    - End of Message (EOM) byte: Indicates the end of a message.
4. Implement code on both microcontrollers to send and receive data according to the defined protocol.

```
#Reciever using C on MSP430 Written by Anthony
#Carthen Sept 20,2020
#include <msp430.h>

void uart_send(char data) {
    while (!(UCA0IFG & UCTXIFG)); // Wait for TX buff-
er to be ready
    UCA0TXBUF = data; // Send data
}

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Configure UART
    P1SEL1 |= BIT2 | BIT3; // UART pins
    P1SEL0 &= ~(BIT2 | BIT3);
    UCA0CTLW0 = UCSWRST; // Reset UART
    UCA0CTLW0 |= UCSSEL__SMCLK; // SMCLK as clock
source
    UCA0BRW = 52; // 9600 baud rate
    UCA0MCTLW |= UCOS16 | UCBRF_1 | 0x4900;
    UCA0CTLW0 &= ~UCSWRST; // Release UART

    // Main loop
    while (1) {
        // Example: Send "impurity" message
        uart_send('H');
        uart_send('e');
        uart_send('l');
        uart_send('l');
        uart_send('o');
        __delay_cycles(1000000); // Delay between msg
    }
}
```

```
#Transmitter using C on MSP430 Written by Anthony
#Carthen Sept 20,2020

#include <msp430.h>

#include <msp430.h>

char uart_receive() {
    while (!(UCA0IFG & UCRXIFG)); // Wait for RX buff-
er to have data
    return UCA0RXBUF; // Return received data
}

int main(void) {
    WDTCTL = WDTPW | WDTHOLD; // Stop watchdog timer

    // Configure UART
    P1SEL1 |= BIT2 | BIT3; // UART pins
    P1SEL0 &= ~(BIT2 | BIT3);
    UCA0CTLW0 = UCSWRST; // Reset UART
    UCA0CTLW0 |= UCSSEL__SMCLK; // SMCLK as clock
source
    UCA0BRW = 52; // 9600 baud rate
    UCA0MCTLW |= UCOS16 | UCBRF_1 | 0x4900;
    UCA0CTLW0 &= ~UCSWRST; // Release UART

    // Main loop
    while (1) {
        // Receive and process incoming messages
        char data = uart_receive();
        // Process received data (e.g., display or use
it)
    }
}
```

Important aspects to consider:

Data Framing: The transmitter sends a Start of Message (SOM) byte (0x02) before the actual message and an End of Message (EOM) byte (0x03) after the message. The receiver waits for the SOM byte before processing incoming data and exits the processing loop when it receives the EOM byte.

Error Handling: The receiver code now waits for the start of a message (0x02) before processing incoming data. This helps synchronize the receiver with the transmitter and ensures that partial or corrupted messages are not processed.

These enhancements improve the robustness of the communication system by providing basic error detection and synchronization mechanisms.

Future Improvements:

There are several aspects missing from the code that would be essential for a complete and robust implementation. Here are some key considerations:

◊ Flow control mechanisms (e.g., hardware flow control using RTS/CTS signals or software-based flow control) are essential for managing data flow between the transmitter and receiver, especially when dealing with large volumes of data or in scenarios where the receiver might be temporarily unable to process incoming data.

◊ Checksum or CRC: Adding checksum or cyclic redundancy check (CRC) calculations to each message can help detect transmission errors and ensure data integrity.

◊ Power Management: The code does not address power management considerations, such as implementing low-power modes or optimizing power consumption for battery-operated devices.

◊ Interrupt Handling: Implementing UART communication using interrupts can improve performance and reduce processor overhead compared to busy-waiting loops.

◊ Synchronization: Ensuring synchronization between the transmitter and receiver is critical for reliable communication. This includes handling synchronization errors that may occur during initialization or data transmission.

◊ Protocol Design: Designing a robust communication protocol tailored to the specific requirements of the application is crucial. This includes defining message formats, command structures, error handling mechanisms, etc.

◊ Testing and Validation: Rigorous testing and validation of the communication system under various operating conditions (e.g., different baud rates, noise levels, and environmental conditions) are necessary to ensure reliability and robustness.

◊ Documentation: Comprehensive documentation, including code comments, interface specifications, and usage guidelines, is essential for facilitating collaboration, troubleshooting, and future maintenance.

Addressing these aspects will contribute to a more complete and reliable implementation of the UART communication system between MSP430 microcontrollers over Bluetooth.

Other uses cases can be embedding the device into a larger system that is submersible into the media being tested. As agriculture becomes and important influence for Americans, devices like this can be implemented in the growing process to monitor contamination, abnormal growths, and bacteria growing within water systems that subsequently is used to grow food.

As this project wraps, I intended to integrate this technology into a larger system, as the principles and technologies leveraged will affect the agricultural community positively.