#include <xc.h>

// Configuration bits

#pragma config FOSC = XT // Oscillator Selection (XT oscillator)

#pragma config WDTE = OFF // Watchdog Timer Enable (WDT disabled)

#pragma config PWRTE = ON // Power-up Timer Enable (PWRT enabled)

#pragma config BOREN = ON // Brown-out Reset Enable (BOR enabled)

#pragma config LVP = OFF // Low-Voltage Programming Disable (RB3/PGM pin has PGM function)

#pragma config CPD = OFF // Data EEPROM Memory Code Protection (Data EEPROM code protection off)

#pragma config WRT = OFF // Flash Program Memory Write Enable (Write protection off)

#pragma config CP = OFF // Flash Program Memory Code Protection (Code protection off)

// Function prototypes

void initUART(void);

void UART\_sendChar(char ch);

void UART\_sendString(const char\* str);

void delay(unsigned int ms);

void main(void) {

// Initialize UART and other necessary peripherals

initUART();

// Send a message to the laptop

UART\_sendString("Hello, Laptop!\n");

while (1) {

// Your main loop code

// Example: Send a command to the camera if needed

UART\_sendString("Capture\n");

// Wait for a while

delay(1000);

}

}

void initUART(void) {

// Configure the UART module

TXSTA = 0x20; // TXEN = 1, BRGH = 0 (Low Baud Rate)

RCSTA = 0x90; // SPEN = 1, CREN = 1 (Serial Port Enabled)

SPBRG = 25; // Baud rate 9600 with 4MHz clock

}

void UART\_sendChar(char ch) {

while (!TXIF); // Wait for TX buffer to be empty

TXREG = ch; // Transmit character

}

void UART\_sendString(const char\* str) {

while (\*str) {

UART\_sendChar(\*str++);

}

}

void delay(unsigned int ms) {

while (ms--) {

\_\_delay\_ms(1); // Requires the correct FOSC setting

}

}