# Stock management with HX711 and UART Communication

**INTRODUCTION:**

This code is designed for the STM32F4 microcontroller to measure weight using the HX711 load cell amplifier and communicate the results via UART. The program initializes and configures the HX711 for load measurement, enabling it to read 24-bit data values that represent the load cell’s analog weight reading. After converting this raw reading into a weight value, the code applies a calibration factor to get a real-world weight measurement in kilograms.

The code also establishes a UART communication link at a 9600 baud rate, allowing the STM32F4 to send weight data to a connected terminal or display. Based on the weight value, feedback is provided via two LEDs connected to GPIO pins on the microcontroller: a green LED indicates sufficient stock, while a red LED signals low or out-of-stock status. Additionally, certain UART commands can trigger specific feedback, such as the difference between a reference weight and the current weight.

For fault detection and debugging, the code includes exception handlers for various system faults such as hard faults, memory management faults, bus faults and usage faults which provide diagnostic messages over UART and halt program execution when these issues occur.

In essence, this program creates a basic weight-measurement system with real-time feedback and diagnostic capabilities, suitable for inventory or stock monitoring applications where weight can determine availability.

**SYSTEM COMPONENETS:**

1. **STM32F4 Microcontroller**: Acts as the central processing unit for the weight measurement system.

**Function**: Manages data acquisition from the HX711, handles UART communication, and controls LED indicators. It executes the main program logic in an infinite loop, continuously checking for weight data and user commands

1. **HX711 Load Cell Amplifier**: Converts the analog signal from the load cell into a digital format that the STM32F4 can process.

**Function**: The microcontroller sends clock signals to the HX711 to read the 24-bit weight data.

**Calibration**: The raw data is converted into a calibrated weight in kilograms using the defined SCALE\_FACTOR and OFFSET constants.

1. **UART Interface**: Facilitates communication between the STM32F4 and an external terminal.

**Data Transmission**: The microcontroller sends formatted weight data and status messages to the terminal at a baud rate of 9600.The system listens for commands sent from the terminal (e.g., to calculate the offset), allowing for user interaction and control.

1. **LED Indicators**: Provide visual feedback on stock levels based on the measured weight.

**Green LED (PA5)**: Lights up when the weight is above the predefined threshold, indicating sufficient stock.

**Red LED (PA6)**: Lights up when the weight falls below the threshold, indicating low stock levels.

1. **Interrupt Handling:** Ensures timely updates from the HX711 when new weight data is available.

**Functionality**: The microcontroller is configured to trigger an interrupt on the data pin (PB7). When a new measurement is ready, it reads the data from the HX711, allowing for immediate processing and display.

**CODE STRUCTURE:**

1. **Constants and Macros:**
2. Pin Definitions:

#define HX711\_DT\_PIN 8

#define HX711\_SCK\_PIN 9

#define HX711\_UPDATE\_PIN 7.

1. Calibri constants:

#define SCALE\_FACTOR 0.000000003

#define OFFSET 8392000.

1. UART configuration:

#define UART\_BAUDRATE 9600.

1. Global Variables:

volatile uint32\_t count=0; : A volatile variable to hold the count of clock cycles, which will be used for reading data from the HX711.

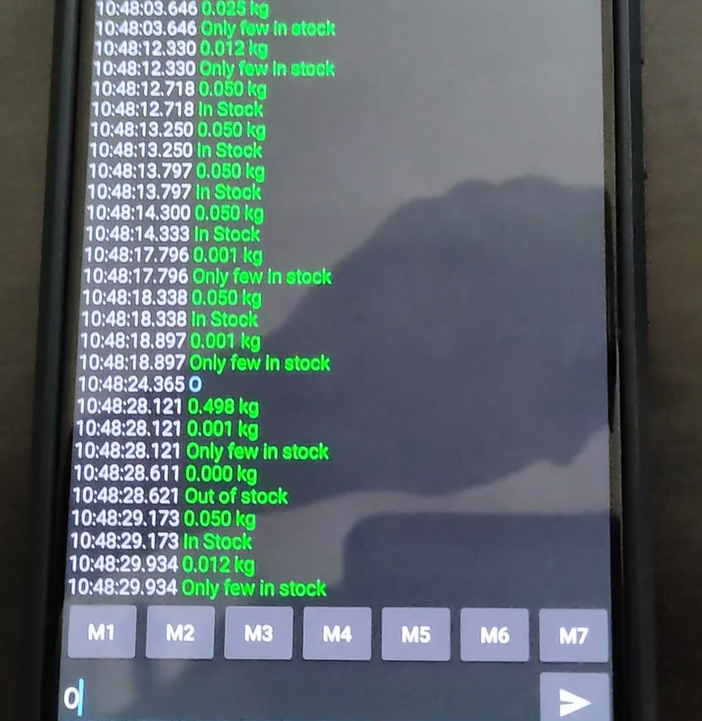
1. **UART Functions:**
2. void UART\_ Init(void): Initializes the UART interface for communication by configuring GPIO pins and setting the baud rate.
3. void UART\_ Send\_ String(char\* str): Sends a string over UART.
4. char UART\_ Receive(void): Receives a character from UART.
5. **HX711 Functions:**
6. void HX711\_Init(void): Initializes the HX711 by configuring the data and clock pins and setting up the interrupt for weight updates.
7. uint32\_t HX711\_Read(void): Reads a 24-bit value from the HX711 by toggling the clock pin and assembling the data.
8. float Get\_ Weight(void): Converts the raw reading from the HX711 into a weight value using the scaling factor.
9. **String Manipulation Functions:**
10. void Float\_ To\_ String (float value, char\* str): Converts a float value (weight) into a string representation suitable for UART transmission.
11. **Interrupt Handlers:**
12. void EXTI9\_5\_IRQHandler(void): Handles external interrupts for the weight update signal from the HX711.
13. void HardFault\_ Handler(void), void MemManage\_ Handler(void), void BusFault\_ Handler(void), void UsageFault\_ Handler(void): Exception handlers that respond to different fault conditions, typically sending error messages over UART.
14. **Fault trigger Functions:**
    * + 1. Functions like Trigger\_ Reset(void), Trigger\_ HardFault(void), etc., are provided to simulate different fault conditions for testing purposes.

**OUTPUT SCREENSHOT:**

DEMO MODEL:



O COMMAND REPRESENTATION (THE WEIGHT TO BE ORDERED):



**CHALLENGES FACED:**

* **Initial Configuration of Hardware Components:** Configuring the GPIO pins correctly for the HX711 and UART communication proved challenging, particularly ensuring the correct alternate functions were set and that the pins were operating as expected.
* **Interfacing with HX711**: Accurately reading and interpreting the 24-bit data from the HX711 required precise timing and synchronization of the clock signal
* **UART Communication**: Ensuring reliable UART communication for sending and receiving data, particularly in terms of handling data transmission errors and ensuring that the transmitted data is formatted correctly.
* **LED Indicator Logic**: Designing a logical state machine to control LED indicators based on weight thresholds effectively.
* **Interrupt Handling**: Managing external interrupts from the HX711 to update weight readings without causing delays or losing data
* **Performance Optimization**: Achieving optimal performance and responsiveness while continuously reading weight data and responding to UART commands.

**SOLUTIONS IMPLEMENTED:**

**SUMMARY:**

**In conclusion, the weight measurement system utilizing the STM32F4 microcontroller and HX711 load cell amplifier exemplifies an effective integration of embedded systems, sensor technology, and real-time data processing. Through careful design and implementation, the system not only meets the functional requirements of accurate weight measurement but also enhances user interaction through UART communication and visual indicators.**

**Future work could involve refining calibration techniques, expanding communication capabilities (e.g., using wireless modules), or integrating additional sensors to enhance functionality. Overall, this project highlights the significant potential of microcontroller-based systems in various domains, paving the way for innovative solutions in real-world applications.**