General Purpose Data Acquisition System Using STM32 Microcontroller and FreeRTOS

E&CE | Department of Electronic and Computer Engineering

Qinyuan Liu

BE Electronic and Computer Engineering

Introduction

The rapid advancement of embedded systems has driven the need for precise, lightweight, and agile data acquisition, especially in fields such as industrial automation, robotics, and control systems. This project focuses on the development of a general-purpose data acquisition system utilizing the STM32 microcontroller and FreeRTOS, providing both analog and digital I/O capabilities with real-time communication functionality.

Applications for this system include battery management, vibration analysis in industrial machinery, and other real-time monitoring scenarios. By leveraging the power of RTOS and embedded design, this project aims to create a scalable and efficient data acquisition platform suitable for various engineering applications.

Aim

This project aims to explore the feasibility of developing a general-purpose data acquisition system using an STM32 microcontroller and FreeRTOS. The focus is on designing and testing a basic prototype that integrates both analog and digital inputs/outputs while evaluating its potential for real-time data processing.

Method

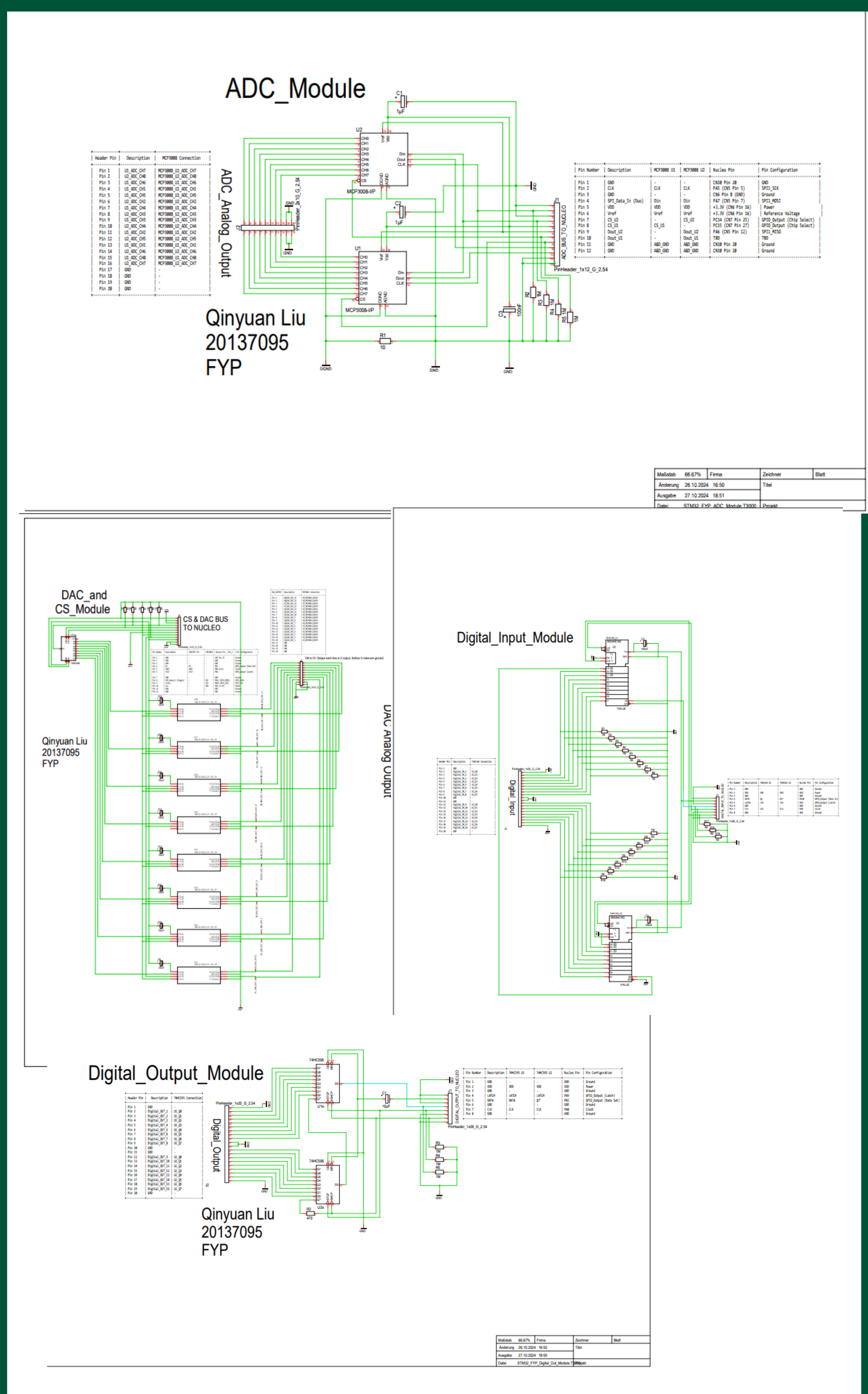
This project investigates the feasibility of a real-time data acquisition system using FreeRTOS and STM32. The methodology follows an iterative experimental approach, with a focus on hardware design, communication protocols, and scheduling strategies.

1.Hardware Setup:

1. The system is built around an STM32 microcontroller.
2. Interfacing external ADCs and DACs (e.g., MCP3008, MCP4822) for future exploration.
3. Digital I/O expansion strategies, such as multiplexing and shift registers, are under evaluation.

2.Software Development:

1. FreeRTOS is used for real-time scheduling and task management.
2. SPI and I2C communication protocols are being explored for peripheral interfacing.
3. A PC-based script is planned to facilitate data visualization and system debugging.



Results

This project is an exploratory study on implementing a general-purpose data acquisition system using STM32 and FreeRTOS. While full SPI communication and is set for future exploration, significant groundwork has been laid in the following areas:

1. System Design: Successfully developed and structured the hardware schematics for 16x16 digital I/O and 16x16 analog I/O.
2. Task Scheduling: Designed a FreeRTOS-based communication function to scan all connected chips at a 1kHz frequency.
3. Hardware Prototyping: Initial integration of MCP3008 (ADC) and MCP4822 (DAC) into the system, with future work needed for proper interfacing.

Full Repository can be found in the link on the left

https://github.com/Qinyuan72/FYP/tree/master

Acknowledgements

Much thanks to Ian for his great support and patience in the overall development, and spatial thanks to my dad in helping implementing the Schematics into actual PCB board

Software

The software component of the project is built using STM32CubeIDE, which provides a comprehensive development environment for STM32 microcontrollers. CMSIS-RTOS V2 is used as the RTOS platform for task scheduling and synchronization. Key features of the software design include:

•Integration of HAL (Hardware Abstraction Layer) functions for simplified interaction with peripherals.

•Task management and communication control using CMSIS-RTOS V2.

Hardware

The hardware component is developed on the NUCLEO-L476RG development board. Key components include:

•**STM32 Microcontroller:** The core processing unit of the system.

•**Analog/Digital I/O:** 16-channel ADC/DAC integration and 16x16 digital signal I/O.

•**PCB Layout:** Designed using Target 3001 for efficient integration and connectivity.

•**Shift Registers and Parallel-to-Serial Converters:** For managing digital I/O efficiently.

The breadboard prototype serves as the initial testing platform, followed by the PCB prototype for advanced design.

Conclusion and personal reflection

Embedded systems are incredibly challenging, requiring deep knowledge across hardware design, real-time systems, and low-level programming. Throughout this project, I encountered numerous difficulties, from SPI implementation to FreeRTOS scheduling, proving that embedded development is not just about writing code but about understanding the intricate dance between software and hardware. One key takeaway from this experience is scope management—as the saying goes, don't chew a chunk you can't swallow. Ambition is great, but practical constraints, time, and personal well-being matter just as much. Breaking down problems into manageable pieces and focusing on incremental progress is crucial in a field where complexity can quickly become overwhelming. Despite the setbacks, this project reinforced my appreciation for embedded systems and their real-world impact. There's still a long way to go, but every failure is a step toward mastery.

UNIVERSITY OF LIMERICK
OLLSCOIL LUIMNIGH