System Block Diagrams

Project FishWorks / ECET 292 / 10.13.2024

Sebastien, Eric, Kayleb and Braden

Table of Contents

Table of Tables

Table of Figures

Cover Imagery Produced by OpenAI

Table 1 FishWorks PCB instruction Manual Revision History

Date Revision History	Revision
October 13, 2024	0.1

PROJECT DESCRIPTION

We are looking to build an aquarium monitoring solution, allowing consumers to build a system comprised of off the shelf components that can centrally manage all aspects of their fish ecosystem. At the heart of our control system is a modular design allowing for a consumer to purchase only the components they require, while being expandable to allow for growth and new aquariums. The system is centrally controlled by a base station with onboard storage and a display for status, warnings and user controllable options. The base station will connect to other nodes which will control and monitor peripheral devices such as temperature and utilize this information to implement fail-safes such as power control. All data and control will be accessible through a web app allowing for real time monitoring, historical data, direct system control as well as setting user defined limits and alarm values.

Table 1 shows the power budget for single base station, multiple node system. 24V DC will be provided to the base station and used to power itself and all connected nodes. This budget assumes a worst-case scenario where all nodes are drawing their maximum allowed current simultaneously. The per node maximum current is estimated before final firmware has been developed and tested, and therefore is using worst case power draw numbers for the processor. Once final firmware can be tested, a lower per node power budget could likely be used, and therefore more nodes could be allowed and/or a smaller power supply could be used.

Table 2 - Full System Power Budget

Input Power:	24V @ 5A(120W) max – via external AC-DC switching power supply.			
Device	**Quantity(max)**	**Voltage**	**Current - Per Device**	**Current – Total**
Base Station	1	24V	185mA	185mA
Node Controller & Attached Peripherals	20	24V	185mA	3.7A
			Total:	**3.885A**
Notes:	*Absolute maximum current values are used for all devices, actual current will likely be much lower under most circumstances.* *A target max 80% utilization is used for all current ratings unless otherwise specified*			

NODE CONTROLLER

The purpose of the node controller is to interface various peripheral devices (sensors, power control, etc.) with the CAN Bus network, allowing data to be sent to/received from the web app via the base station. Each node receives 24V power from the CAN Bus network. The power budget (Table 3) for the node includes both the node controller itself and any power required by the attached peripheral devices.

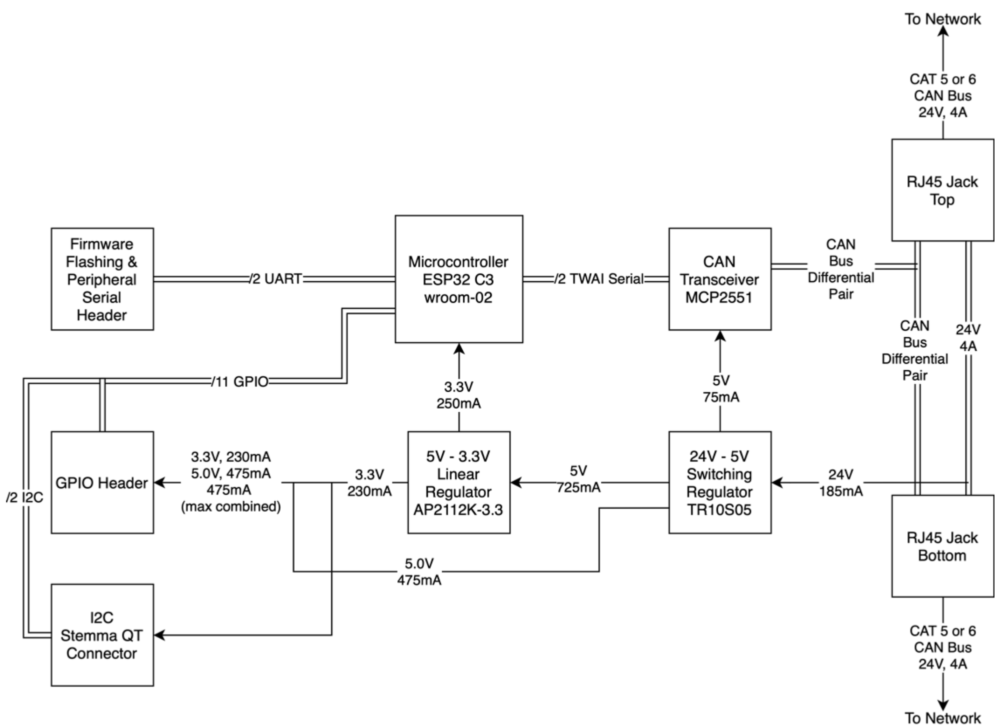

Figure 1 - Node Controller Block Diagram

Table 3– Node Controller Pins

Block	ESP32 C3 GPIO	Protocol	Config	Description
Stemma QT Connector	SDA=10, SCL=5	I2C	Dedicated Peripheral	Compatible various I2C connector standards
GPIO Header	10,9,8,5,4,18[1],19[1],3,2,1,0, TXD[2], RXD[2]		GPIO	GPIO for connected device. UART programming pins broken our if needed.
CAN Transceiver	TX=6, RX=7	TWAI	Dedicated Peripheral	Connection to CAN Bus
UART Programming Header	TXD, RXD	UART	Dedicated Peripheral	For firmware flashing
Notes:	*1 – GPIO 18 and 19 are configured for USB/JTAG by default, they must be reconfigured to be used as GPIO.* *2 – TXD, RXD must be available to be used for firmware flashing.*			

Table 4– Node Controller Power Budget

Input Power:	24V @ 185mA max – from base station via CAT 5 or 6		
Device	**Voltage**	**Current**	**Source**
CAN Transceiver	5.0V	75mA	24V – 5V switching regulator [2]
Microcontroller	3.3V	250mA	5V – 3.3V linear regulator
GPIO – 5.0V Output	5.0V	475mA[1]	24V – 5V switching regulator[2]
GPIO – 3.3V Output	3.3V	230mA[1]	5V – 3.3V linear regulator
		Total:	**800mA @ 5.0V[2]** **185mA @ 24V[2]**
Notes:	*1. GPIO 5.0V and GPIO 3.3V outputs must not exceed a combined 475mA.* 2. Assuming a 90% average switching regulator efficiency. *Absolute maximum current values are used for all devices, actual current will likely be much lower under most circumstances.* *A target max 80% utilization is used for all current ratings unless otherwise specified.*		

BASE STATION

The purpose of the base station is to send/receive CAN Bus message from attached nodes and send/receive these messages to the MQTT broker to allow the web app to interface with the system. The base station also provides 24V power to all connected nodes. As well the base station provides a basic user interface for the system, notifying the user of alerts from the system via a buzzer and notification LEDs. The user will be able to view details and silence these alerts using the user buttons and TFT screen.

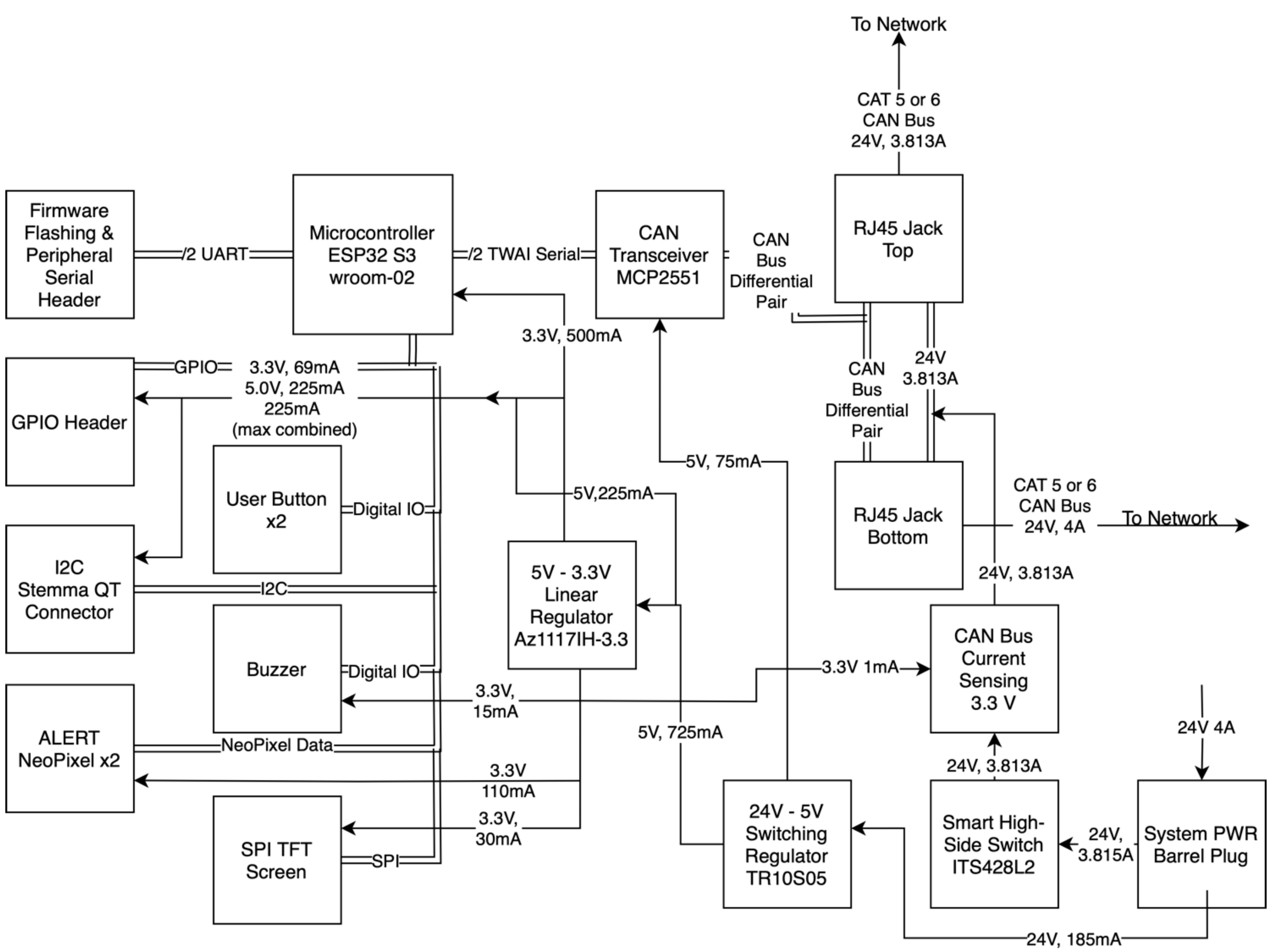

Figure 2 Base station block diagram

Table 5 - Base Station Pins

Block	ESP32 C3 GPIO	Protocol	Config	Description
Stemma QT Connector	SDA=14, SCL=13	I2C	Stemma QT Connector	Compatible various I2C connector standards
GPIO Header	1,2,3,8,18,19,20,21,40,41,42, 43(TXD)[2],44(RXD)[2],45,47,48		Header Pins, GPIO	GPIO for connected devices. UART programming pins broken out if needed.
SD Card	4(CD),5(DAT1),6(DAT0),7(CLK), 15(CMD),16(DAT3),17(DAT2)		PCB Routed GPIO	Dedicated GPIO for SD Card
CAN Power Switch	IN=11, STS=12		PCB Routed GPIO	Dedicated GPIO for CAN Bus power switching
CAN Current Sense	SDA=9, SCL=10	I2C	PCB Routed I2C	Dedicated GPIO for CAN Bus current switching
CAN Transceiver	TX=39, RX=38	TWAI	PCB Routed, Dedicated Peripheral	Connection to CAN Bus
UART Programming Header	TXD(43),RXD(44)	UART	PCB Routed, Dedicated Peripheral	For firmware flashing
Notes:	*1 – GPIO 18 and 19 are configured for USB/JTAG by default, they must be reconfigured to be used as GPIO.* *2 – TXD, RXD must be available to be used for firmware flashing.*			

Table 6 - Base Station Power Budget

Input Power:	24V @ 5A max – from external AC-DC power supply		
Device	**Voltage**	**Current**	**Source**
NeoPixel x2	5.0V	110mA	24V – 5V switching regulator [2]
LCD Screen	3.3V	30mA	5V – 3.3V linear regulator
ESP32 S3	3.3V	500mA	5V – 3.3V linear regulator
Buzzer	3.3V	15mA	5V – 3.3V linear regulator
CAN Bus Power Switch	24V	1.5mA	CAN Bus 24V, external AC-DC power supply
CAN Bus Current Sense	3.3V	1mA	5V – 3.3V linear regulator
CAN Transceiver	5.0V	75mA	24V – 5V switching regulator [2]
Microcontroller	3.3V	500mA	5V – 3.3V linear regulator
Peripherals (including I2C)	5.0V	225mA[1]	24V – 5V switching regulator[2]
Peripherals (including I2C)	3.3V	69mA[1]	5V – 3.3V linear regulator
		Total:	**800mA @ 5.0V[2]** **185mA @ 24V[2]**
Notes:	1. Peripherals *5.0V and 3.3V outputs must not exceed a combined 225mA.* 2. Assuming a 90% average switching regulator efficiency. *Absolute maximum current values are used for all devices, actual current will likely be much lower under most circumstances.* *A target max 80% utilization is used for all current ratings unless otherwise specified.*		

SOFTWARE

Software Overview

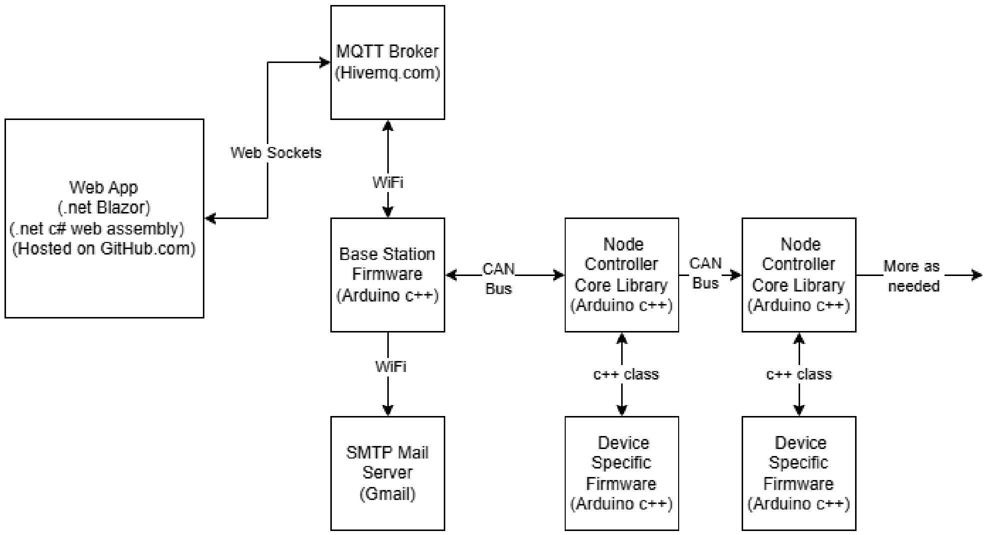

Figure 3 - Software Block Diagram

Node Controller

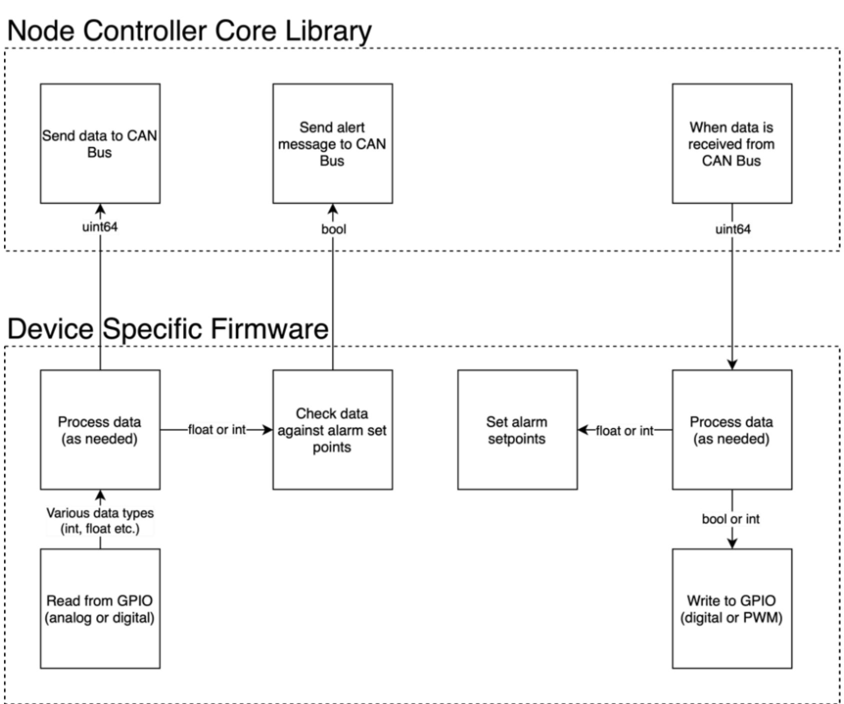

Figure 4 - Node controller software flow chart

Base Station

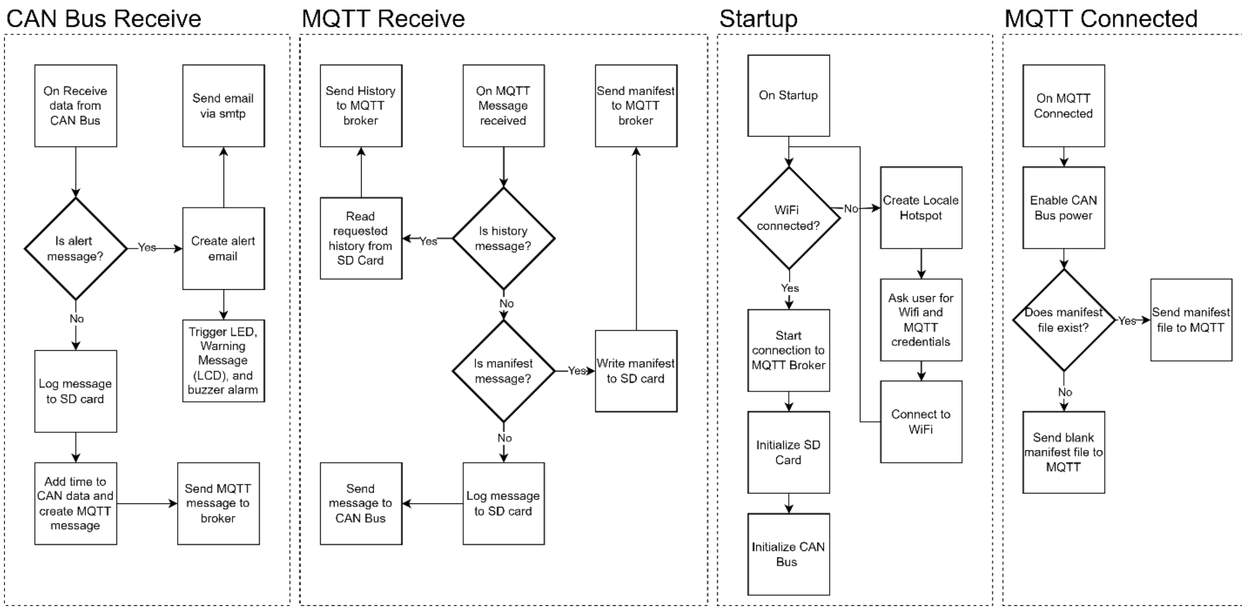

Figure 5 - Base station software flow chart

Web App

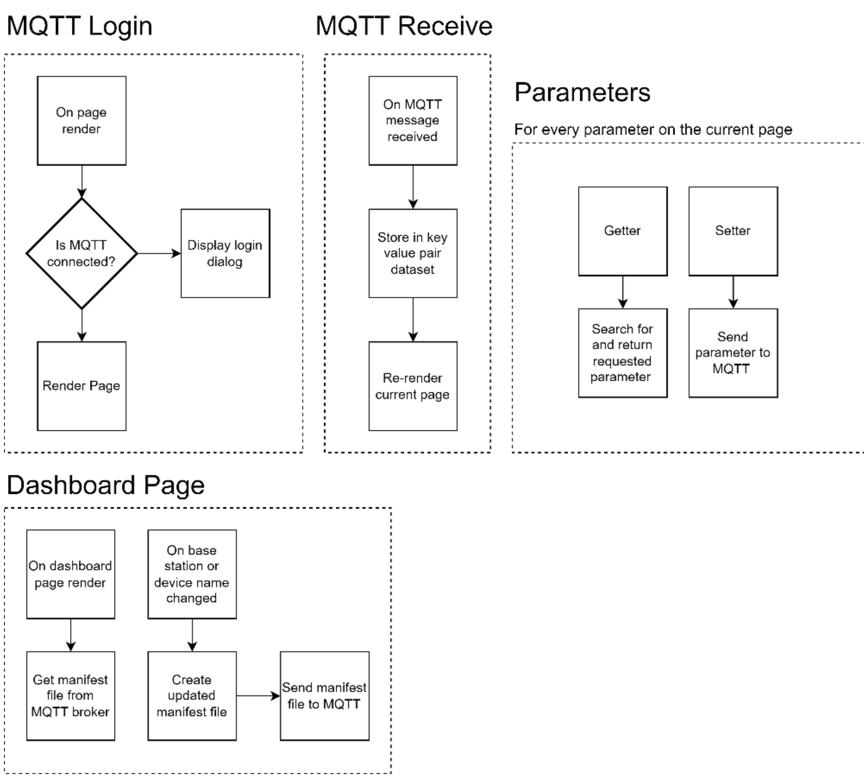

Figure 6 - Web app flow chart