

Aquaponic Microcontroller System Proposal

Jonah Bufford and Cypress Payne



Introduction

Aquaponics is becoming more and more relevant as it has proven to be a highly sustainable, environmentally friendly form of agriculture. A single aquaponic tank can support multiple species of plant life and fish while avoiding potentially harmful chemicals and fertilizers and saving on water and electricity.

Aquaponics is a plant growth system that combines a hydroponic planter connected by a pump to a tank containing fish or other aquatic life. The pump transfers the water

between the two sections of the system, allowing the plant life to gain nutrients from the fish waste in the water and the water to be cleaned and filtered by the plants before it is returned to the fish. By recycling the water and only pumping water through the system when necessary, the system is very efficient for anyone, from large scale fish and produce farmers to young children learning to care for plants and animals.

The proposed device is an aquaponic plant-watering microcontroller based system. The system will allow a user to automate their aquaponic system by setting a timer or relying on the soil moisture sensor in the device to activate the pump. By automating the watering and lighting system, an aquaponic plant tank becomes even more low-maintenance and stress free while ensuring the plants and fish remain healthy and happy.

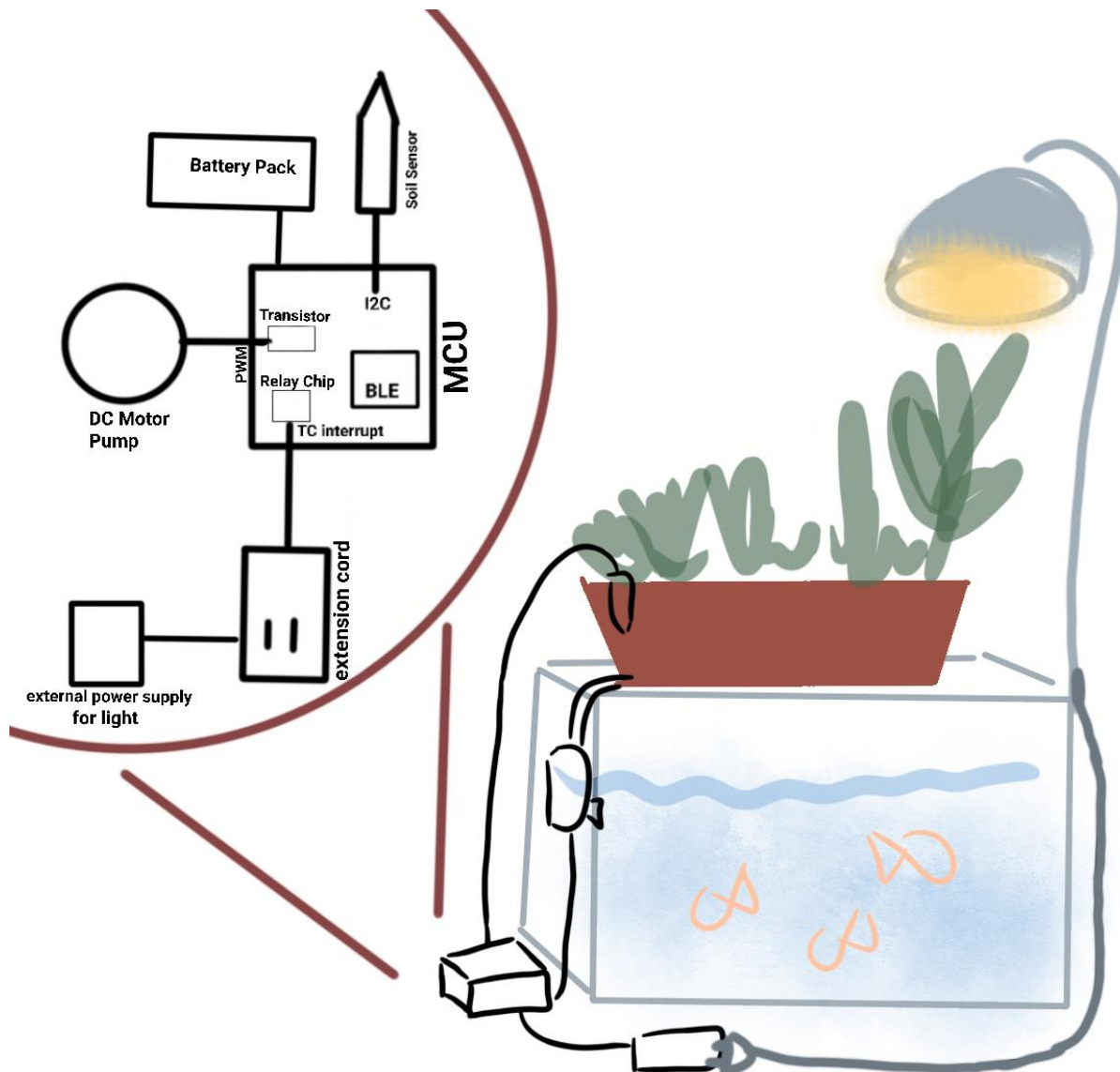
This project is well-suited to a microcontroller-based design because the processing power needed is low. The target market will be folks who are interested in planting and aquaponics, but don't necessarily want to be doing everyday monitoring. A microcontroller-based device will be cheap, easy to install, and appealing to hobbyists who can attach it to a tank, potentially set up some piping, and then move on to a new project while only occasionally needing to check the growth of their plants.

Design Overview

The pump system will consist of a PSoC 4100_BLE microcontroller chip, a pump and a power relay for a grow light as the outputs, and a soil moisture sensor and bluetooth connection to a user interface as the inputs. A user interface in the form of an application will allow the user to set the system's timers and sensor levels to their desired settings. This input will be communicated to the MCU through the bluetooth connection. The software will consist mainly of device drivers for the required hardware and functions for reading and writing to and from the inputs and outputs of the system. Software connections to communicate with the hardware will include an I2C, BLE, and multiple TCPWMs.

The main device will be encased in hard plastic with wire connections to the external hardware. The soil sensor will be placed in the aquaponic tank's planter to get data on the moisture levels. The pump will be connected to both the fish tank and the planter by plastic tubing. The power connector for the light will have a type b plug adapter that a grow light can be plugged into to put the light on a timer controlled by the user.

Fig 1: System Overview



User Interface

A user interface will be set up for setting the timing of the water and lighting. The app will have three functions: allowing the user to set a time and duration for watering, allowing the user to set a moisture level for watering, and allowing the user to set times for the lamp to be on and off. It will also need to allow the user to choose whether to use a real time clock timer or the soil moisture sensor for the pump. Once the user has configured their settings, the app will use bluetooth to connect to the chip, transferring the times and settings to the MCU software.

Connecting to the device through the app will also automatically set the time for the real time clock. The device will have an LED that will shine green to indicate the device is on and working properly. When the power has been turned off, the light will shine red to indicate that the clock has been disrupted and needs to be updated using the app.

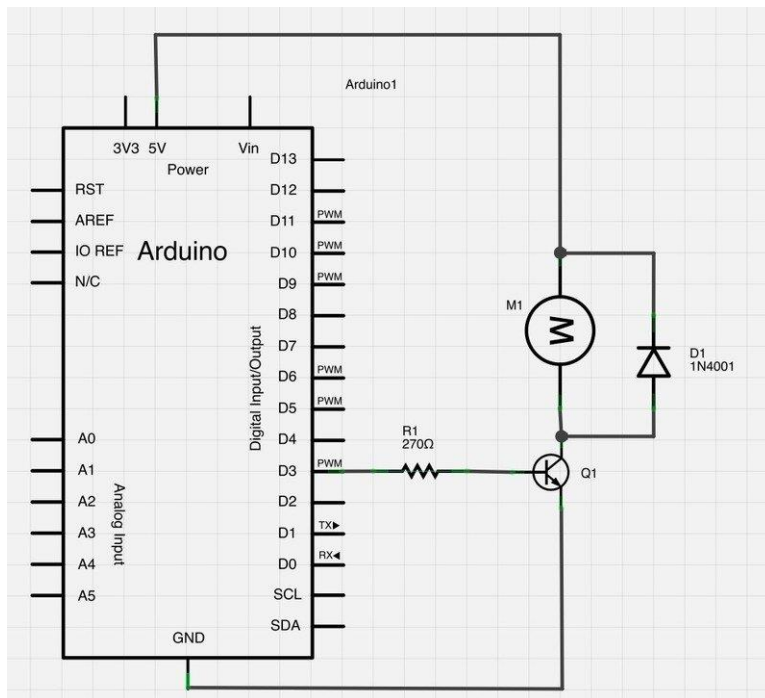
Hardware

This device will be built using the PSoC 4100_BLE chip. The memory component for the program will be relatively low; we expect that the onboard memory should be sufficient.

The only physical input on the device will be the power switch. All other communication will be through the app, using the bluetooth connection. The device will advertise constantly, allowing a user to connect with the app using a phone or computer whenever they want.

One primary hardware component of the device will be a pump. The pump will use a DC motor driven by a digital output pin and controlled with the use of a transistor. The transistor will have three pins, a base, a collector, and an emitter. One wire from the motor will be connected to a 12 V power supply, while the other will be connected to the collector pin of the transistor, and a diode will cross the motor from the collector to the power supply. The transistor's base will be connected to a digital output pin from the PSoC chip, and the emitter will connect to ground.

Fig. 2: Transistor Circuit with Motor (Source: “Transistors”, Adafruit)

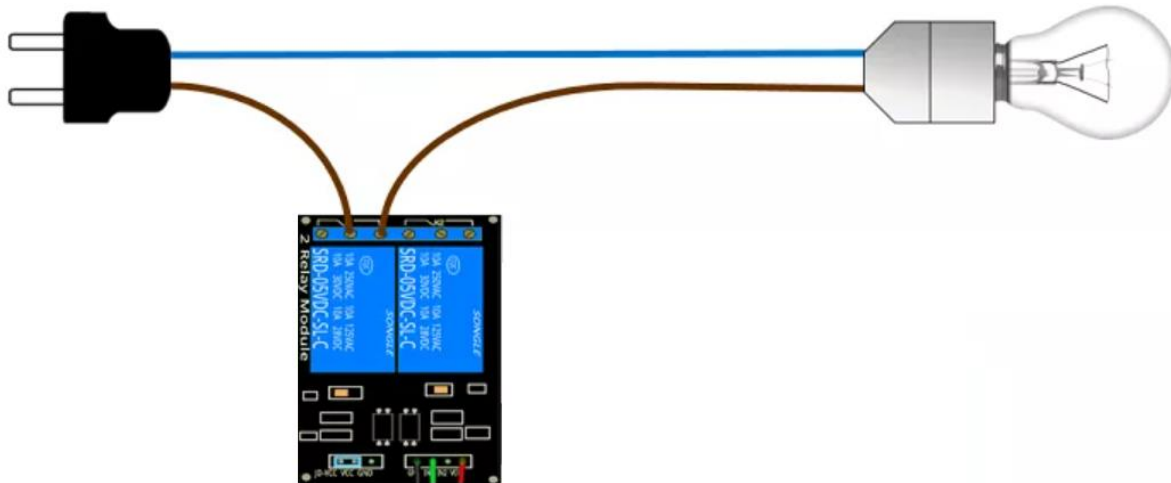


The DC motor will be connected to a plastic impeller. The impeller will spin within the body of the pump, pulling in water and pushing it out through the top. Most of the pump will be contained within the device casing. Two two-foot long bendable plastic tubes will extend from the device; one will pull water from a water supply, and the other will expel water. This tubing could be attached to a more complex piping system by the device owner depending on the needs of their setup.

The power control for a grow lamp will be implemented using an extension cord that will be connected to a relay chip by 2 feet of wiring. The extension cord will need to be plugged into an external power source as well as the grow lamp (this could be also used to power any electronic device on a timer). The relay chip will act as a switch controlled by the software. The relay chip will have a wire attached to an output pin on the PSoC chip, as well as wires attached to the 5V power source and ground. In the extension cord, one of the wires will be split; each end will be connected, using added wire, to a terminal in the relay chip. The relay chip has three terminals for wires; a “common terminal” and a normally open/normally closed terminal. The wires connected to the extension cord will be soldered

to the common terminal and the normally closed terminal, meaning power will be off unless the relay chip gets a “high” signal from the input.

Fig. 3: Power Control Cable Attached to Relay Chip (“Guide For Relay Chip With Arduino”, Random Nerd Tutorials)



The device will also include a soil moisture sensor. The moisture sensor will be purchased from Adafruit and connected to the device by 2 feet of insulated wire. The program will communicate with the moisture sensor using I2C, and can use moisture levels in the soil to determine watering times instead of a timer. The sensor will need four wires attached; one connected to an SDA pin, one to SCL, and then a wire for connecting to power and ground. The sensor requires 3-5 volts of power, meaning it can use the same power supply as the chip.

The PSoC chip, relay chip, H-Bridge chip, motor, and pump body will all be contained in a plastic casing. The H-Bridge and relay chips, as well as the LED, will be connected to the PSoC chip with soldered wiring, as will the moisture sensor if included. The motor will be connected to the H-Bridge chip by soldered wires, and the power control will be connected to the relay chip with soldered wires.

The device will be powered by a battery pack on the back. While all the batteries will be stored in the same place, there will be two separate power supplies; a 12 volt supply for the motor, and a 5 volt supply for the other components. The other compnet

One side (the “front”) of the casing will have an led light (described in user interface), a power switch, and the port for the intake tube of the pump. The output tube for the pump will come through the “top” of the device. The insulated wire connecting to the moisture sensor will also come out the top, and the power connector will have a wire pushing through the left side.

The device will have a real-time clock configured with an internal timer, which will trigger the watering (if the user is using time-based watering). The power for the device should remain on all the time when it’s in use so that this clock can run. If the device is turned off and turned back on, the LED will glow red until the device is reconnected to the app through a bluetooth connection; the app will automatically set the device time to the correct time.

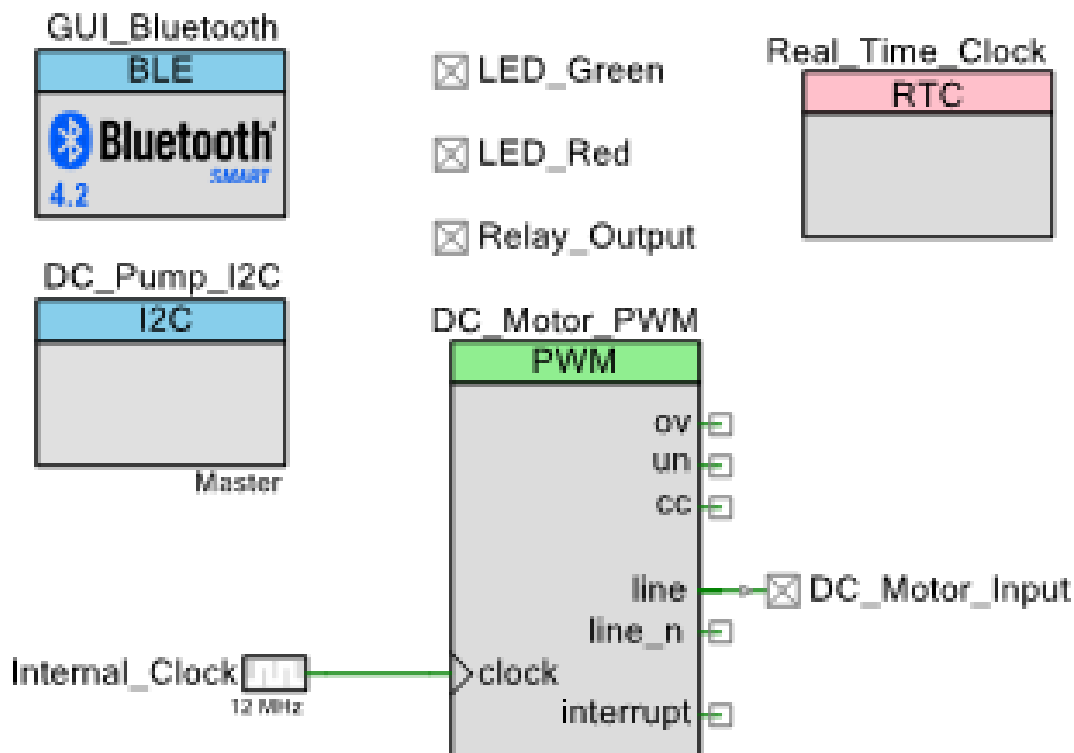
Software

The main software requirements will involve interfacing with the various inputs and communicating with the pump and the grow-light (if the grow-light connection add-on is selected). The two inputs that will be involved are the Adafruit Soil Sensor and a bluetooth chip for communication from the user. The outputs are the power relay for a grow light connection and pump.

The Adafruit Soil Sensor uses an I2C connection for communication between the microcontroller and the sensor itself. The bluetooth chip is built into the PSoC MCU and will require a BLE software connection. The relay chip will require an output pin connected to a timer-counter that will signal when to turn the light on or off. A PWM will be used to control the pump, which is built using a DC motor. A real time clock will be necessary for determining when the timer-counter and PWM interrupts will turn on the pump and the

grow light respectively. Two other output pins will also be included for the LED light in green and red, which will communicate to the user the state of the real time clock. Device drivers will be necessary for the aforementioned input and output connections. Basic programming will then cover the interrupts and a simple main function with startup code and a loop.

Fig 4: Top-Design Schematic



When the app connects to the device, the app will set a few variables. It will set the time for the real time clock, a list of times for the lamp power to turn on and off, a list of times for the pump to turn on and off, and a moisture range for the pump to run. The app will also be able to enable or disable any of these options. The main loop will sample moisture and

check time, and if the time or moisture level meets the conditions, it will call the associated function to turn on the appropriate component. Turning on the lamp will just require setting the digital output pin for the relay chip to high, and turning it off will require setting it low. Turning on the DC motor for the pump will require setting the compare time for the associated PWM, and turning it off will require setting the compare time back to 0.

The size of the software itself is expected to easily fit on the 256 KB of flash available on the PSoC 4100_BLE MCU and will be programmed using the PSoC Development environment in C.

Cost Estimate

The total cost of a single unit is estimated to be about \$39.65 total. This breaks down into five basic parts. The PSoC 4100_BLE is priced at \$5.15 and the Adafruit STEMMA Soil Sensor at \$7.50. The pump is compiled of a DC motor, H-Bridge transistor, and a plastic impeller, body, and tubes which is estimated to cost around \$7.00 total. The power relay which will control the grow-light is broken up into the relay chip, an extension cord, wire, and insulation and is estimated to total at \$8.00. Finally, adding a generous budget of about \$12 for miscellaneous parts, the estimated final price per unit is calculated to be \$39.65.

Fig 5: Price Per Unit

Component	Price
PSOC 4100_BLE	\$5.15
Pump	\$7.00
Adafruit STEMMA Soil Sensor	\$7.50
Power Relay	\$8.00
Miscellaneous Parts	\$12.00
Total	\$39.65

Retail Price	\$80.00
Expected Profit	\$40.35

The PSoC Development environment for the software programming will require a one-time investment purchase of \$75, but this will easily be made back during sales of the product.

The system itself is expected to sell for a retail price of \$80 dollars.

Conclusion

A microcontroller system to control an aquaponic pump and grow-light increases the efficiency and sustainability of an aquaponic tank. With minimal processing power required, a microcontroller is the perfect solution to ensuring the health of an aquaponic system and allows the buyer to enjoy the benefits of an aquaponic tank with very little required maintenance. The proposed design uses cost-effective hardware and reliable software to create a user-friendly, automated system that will keep both the buyer and their plants happy.