Prototype Demo

Beverly Yee & Alex Gray

What did you build?

The prototype built is a small water pump that's hooked up to a Discovery STM32f072rb board. The two components are hooked up to a MOSFET, 2 1k Ohm resistors, a diode, and a potentiometer (part 3362p, 3362 – 1/4″ Square Trimpot® Trimming Potentiometer) to simulate a varying input.

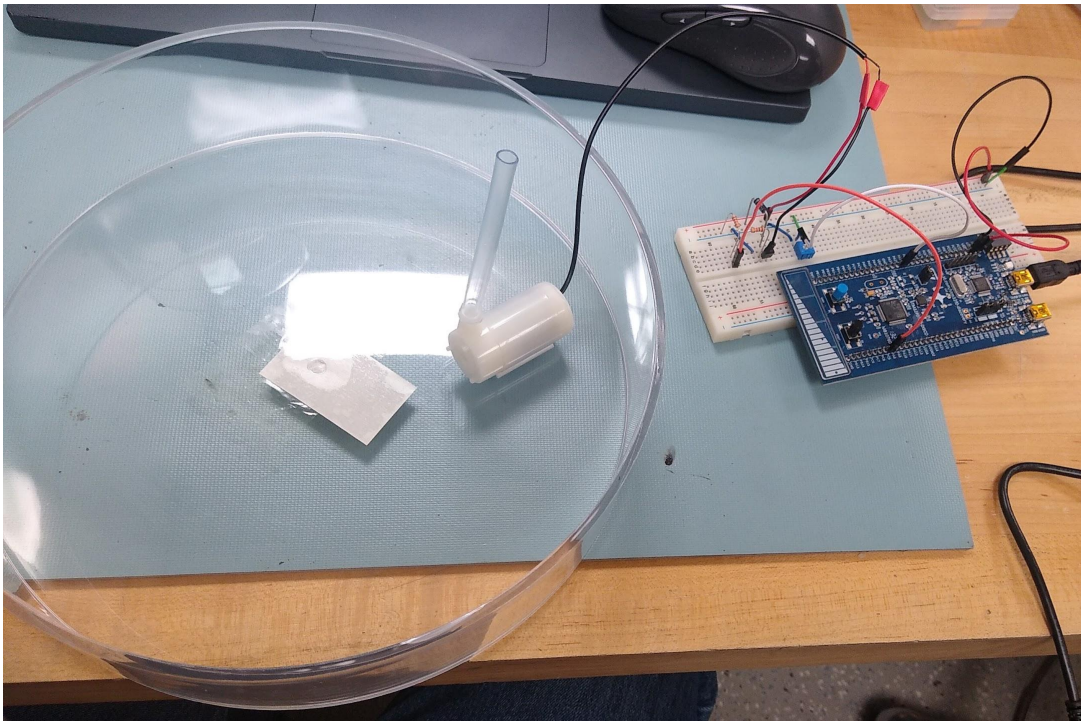

Figure 1. Picture of the complete prototype, without the water.

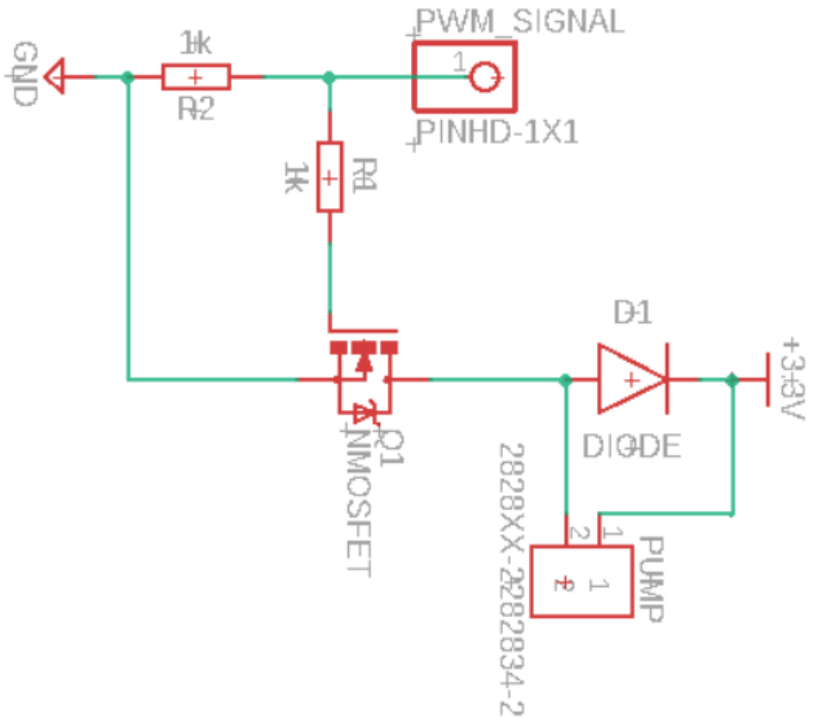

Figure 2 - Circuit schematic of the prototype.

The first thing done was to make sure the circuit is working the way it's supposed to. To that end, a green LED was used in place for the pump. The only difference between the test circuit and the final prototype circuit is the output and an additional wire to power the LED. The PWM used is one Alex created for his Embedded Systems project with a few minor tweaks.

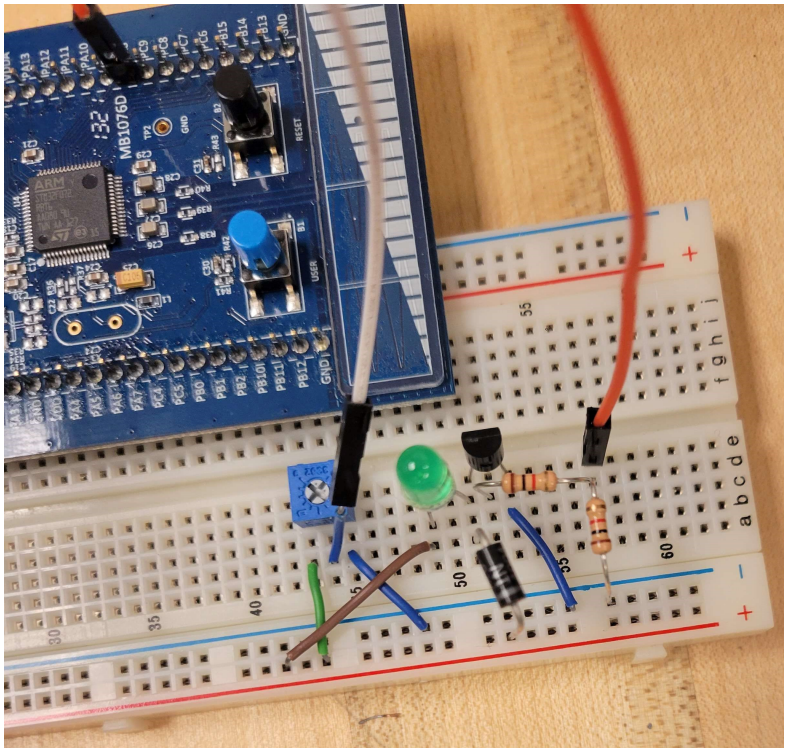

Figure 3. LED circuit

What did you learn?

Initially, the circuit worked as anticipated with the potentiometer controlling the brightness of the LED as anticipated. But when the pump was substituted in, it acted as though there was not enough power going through it. The motor was faintly heard and water was definitely not being pulled through the pump.

Initial analysis made it seem as though perhaps the MOSFET had burnt out through the test circuit. There was also a mishap where STMCube decided to erase previously made code and some of the settings had been reset, such as the ADC not reading continuously to get the ever-changing value of the potentiometer. The LED is still being lit and controlled as before, so the issue was not with the rest of the circuit.

Afterwards, the pump was checked separately to make sure it operated as expected. By itself, plugged into a 5V source and GND, it works great. The problem didn't lie within the hardware, but the software.

Going back to it, Alex discovered that the values the PWM was reading were way out of the range of what the potentiometer could output. Even though the potentiometer was fully opened, not all the current/voltage was being pushed through because the PWM was creating a range between 0 and 65535 instead of 0 and 4000.

```
int main(void)
{
  /* Reset of all peripherals, Initializes the Flash interface and
the Systick. */
  HAL_Init();

  /* Configure the system clock */
  SystemClock_Config();

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_I2C2_Init();
  MX_SPI2_Init();
  MX_TSC_Init();
  MX_USB_PCD_Init();
  MX_TIM1_Init();
  MX_ADC_Init();

  uint16_t readvalue;
  HAL_ADC_Start(&hadc);
  HAL_TIM_PWM_Start(&htim1,TIM_CHANNEL_1);

  /* Infinite loop */
  while (1)
  {
     HAL_ADC_PollForConversion(&hadc,1000);
     readvalue=HAL_ADC_GetValue(&hadc);
     __HAL_TIM_SetCompare(&htim1, TIM_CHANNEL_1, readvalue);
  }
}
```

Both the bugs were found within the STMCube program where the ADC was not set to continuous read and the PWM counter was initially set to 65535. A note of it is made here

in case it needs to be referred to again for the final project. It also should be noted the reset button on the Discovery board needs to be pushed each time new changes are made.

What questions did the demo answer and bring up?

The primary question was whether we could get the pump to pump water with varying heights using a PWM. A big part of the project is to do just that in tune with the music. If it's successful with one pump, then the process can simply be replicated to the other pumps.

The next step is to turn the test circuit into a PCB. The question with this step is whether to create 16 individual PCBs for each pump or make one large PCB with the 16 individual circuits routed to their own header pin. The former would make the final circuit look messy while the later would make it look cleaner and more organized.

Question 2: Each pump must be controlled separately as each pump will have a different height dependent on the signal from the PWM. Does the Discovery board have enough channels for them all? Not exactly. Judging from the datasheet, the board has a max of 12 channels and seems to be the only microcontroller to have that many channels.

Question 3: Use an LED strip to line the outside of the container or create individual LED PCBs in a circle with a hole in the center? The former is cleaner to look at and can come already waterproofed. The latter allows each column of water to be accentuated with light, but creates more clutter and wiring headaches.

Question 4: How much power would be required for all the pumps used? Each requires a minimum of 3V and can handle a max of 5V. There's also LEDs to think about as well.

What would you do differently in the final project?

Instead of a circuit with through-hole parts, create a PCB with SMDs, and maybe even add an LED to ensure the circuit is working properly.

A suggestion was made to use a slightly bigger MOSFET. The current MOSFET works for the test circuit, although it is possible that it might not be strong enough for the circuit.

Another suggestion was made to use an LED driver. While the drivers are obviously designed for LEDs, they can be used for servos. If they can be used for servers, then it stands to reason that they can also be used for pumps.

There were 2 LED drivers discussed in class: STP08DP05 and the MAX7219. The former can only drive 8 LEDs per chip, but the chips can be connected together. The latter, on the other hand, uses an array of 8x8 for LEDs. It's a bit more complicated than the first chip. For now, however, the intention is to use the Discovery board's 12 channels for 12 pumps with the remaining 4 as extras in case something happens.