DIY simulator helicopter collective

By **Andy Frey,** Office Chair Pilot

I love all things aeronautical. I've got many friends who own aircraft, so I can fly in real life. Most of the time, however, I fly in virtual reality on Microsoft Flight Simulator.

Lately, I've been enjoying flying helicopters. The problem is, my simulator cockpit setup doesn't include a helicopter collective, which is the lever to the left of the pilot's seat that controls the pitch of the main rotor blades—pull it up to generate more lift, lower it to decrease lift. The upper part of the lever is the throttle handle that the pilot twists to control the engine throttle.

Off-the-shelf or commercial sim helicopter controls can be quite pricey, or not exactly what I want, so I decided to design my own. Ultimately, this project is just a USB game controller. Creating one is reasonably easy with an Arduino-flavor microcontroller capable of acting as a USB HID-class device (keyboard, mouse, or joystick) and an appropriate HID controller C library, where HID stands for "Human Interface Device."

For this project, I'm using the Sparkfun Pro Micro C board based on an Atmel ATmega32U4 microcontroller. This tiny development board can act like a HID device. There are many examples on the web for making custom USB HID devices with an Arduino.

For the collective, I'm using two Osram AS5600 12-bit contactless hall effect encoders: One is for measuring the angle of the collective lever over a 40º arc, and the other is for measuring the twist of the throttle handle (~100º arc). The AS5600 resolution is 4,096 pulses per 360º. The MCU reads the values of the AS5600 via I^{2}C. The ATmega32U4 has only one I^{2}C port and the AS5600 has just a single I^{2}C slave address, so I'm using an Adafruit PCA9546 4-channel I^{2}C multiplexer to communicate with both encoders.

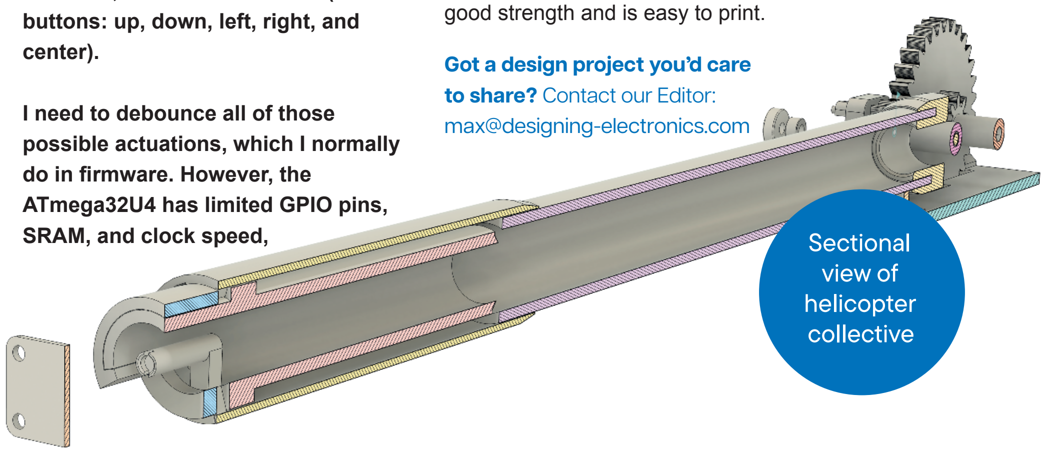
As for the mechanical stuff, I've designed a larger, partial spur gear into the hinged end of the lever which drives a small spur that holds the magnet. The gear ratio increases the resolution of the lever magnetic encoder so that it outputs approximately 4,000 pulses through its 40º arc of rotation. This makes the axis in the simulator more stable, but plenty sensitive to input (gives a smooth but fine touch).

The head of a collective typically has several input controls for various functions. My design has about 27 unique input contacts between four pushbuttons, two (on)-off-(on) paddle switches, two toggle switches, and three POV hats (five buttons: up, down, left, right, and center).

I need to debounce all of those possible actuations, which I normally do in firmware. However, the ATmega32U4 has limited GPIO pins, SRAM, and clock speed, so this timeusing hardware debouncing. This is achieved with three LogiSwitch LS20-P switch debounce ICs (each debounces up to nine switches) and four 74HC165N 8-bit parallel in, serial out shift registers. This makes it ludicrous-easy to track up to 27 switches using only three GPIO pins (clock, serial in, and latch). Every time through the MCU run loop, I shift all the already-debounced switch bits from the shift register and report them through USB as game controller button states. The magnetic encoders are reported as axes.

For designing the mechanical 3D printed stuff, I use Autodesk Fusion 360, which also does PCB design—they integrated Eagle into the app making it easy to design your electronics and build your product around them. I guessed at the length of the collective lever and ordered an aluminum tube on Amazon. The plastic parts are designed to connect to both ends of that tube. I am printing the plastic parts in PETG, which has good strength and is easy to print.

Got a design project you'd care to share? Contact our Editor: max@designing-electronics.com



Sectional view of helicopter collective