

Name	SID	Checkoff

Objectives: Motor and motion control

The application combines the following elements:

- a) DC motors
- b) Optical rotary encoders
- c) H-Bridge
- d) 9 degree-of-freedom inertial measurement unit (9-DOF IMU)

The key to getting all this working is to construct and test each part individually, and then gradually assemble them.

Although simple, the setup demonstrates the capabilities and may serve as a starting point for more sophisticated applications. After completing the lab, you are encouraged to improve the setup (e.g. adjust the range of the joystick control for better sensitivity) and try other ideas. Or hook up a speaker for real sound.

Parts and Tools

HUZZAH32 board, robot kit, encoders, DRV8833 H-bridge, MPU9250 IMU

Prelab

Read the entire instructions.

DRV8833

Study the DRV8833 datasheet and write code to control the direction and speed of a DC-motor.

- Use “slow decay” (see Table 2 in the datasheet)
- Note: PWM duty cycle 100 % corresponds to full speed, 0 % to stop
- Hint: duty cycle 100 % is the same as logic 1. You may use this to control motor direction, i.e. define two PWM channels, one for each motor control pin, and use duty cycle settings to control both speed and direction.
- Frequency: 10 kHz is a reasonable value, experiment in the lab with different values.

MPU9250

Check out the documentation of the micropython driver for the MPU9250 at <https://github.com/micropython-IMU/micropython-mpu9x50>.

Write a program that initializes the MPU9250 and prints the three acceleration, rate, and magnetic field values as well as the temperature once every 200 ms. Replace the argument 'X' in the constructor with an initialized I2C object.

Wiring Diagram

Draw a wiring diagram for all components:

- a) Huzzah32
- b) Battery pack (4 AA, 6 V). This connects to the DRV8833 and, for autonomous operation (i.e. with the USB cable disconnected from the Huzzah32), to VUSB and GND pins of the Huzzah32 board.
- c) DRV8833. See pin diagram at <https://www.pololu.com/product/2130>: connect the battery voltage to VIN, GND, PWM control signals (from the Huzzah32) to AIN1, AIN2, BIN1, BIN2, and the motors to AOUT1, AOUT2, BOUT1, BOUT2. Leave pins VMM, nSLEEP, nFAULT, AISEN, BSEN unconnected.
- d) DC motors.
- e) Power switch, if your motor kit includes one. It goes in series with the battery pack.
- f) Rotary encoders. Note: the pin labelled 5 V goes to the 3.3 V output of the Huzzah32 board. **Do not connect to 5 V, as doing so results in 5 V at the OUT pin and will destroy the Huzzah32 when connected to this pin.** Connect OUT to an input of the Huzzah32.
- g) MPU9250. See diagram at <https://www.sparkfun.com/products/13762>. Connect only GND, VDD, SDA, SCL. VDD goes to the 3.3 V output of the Huzzah32.



Update Firmware

If you have flashed the ESP32 before March 17, 2018, reflash the firmware. A filter option was added to the pulse counter driver described at https://github.com/bboser/IoT49/blob/master/doc/digital_io.md#encoder.

Prelab Checkoff

At the start of the lab session, show the following results from the prelab to the instructor to get credit.

Task	Points	Checkoff
DRV8833 code	10	
MPU9250 code	10	
Wiring Diagram	10	

Lab

Assemble the Robot

Assemble the robot. There are no instructions—use your imagination and Figure 1 as a guide. The solderless protoboard and battery pack (on the underside of the chassis) can be attached semi-permanently with Velcro.

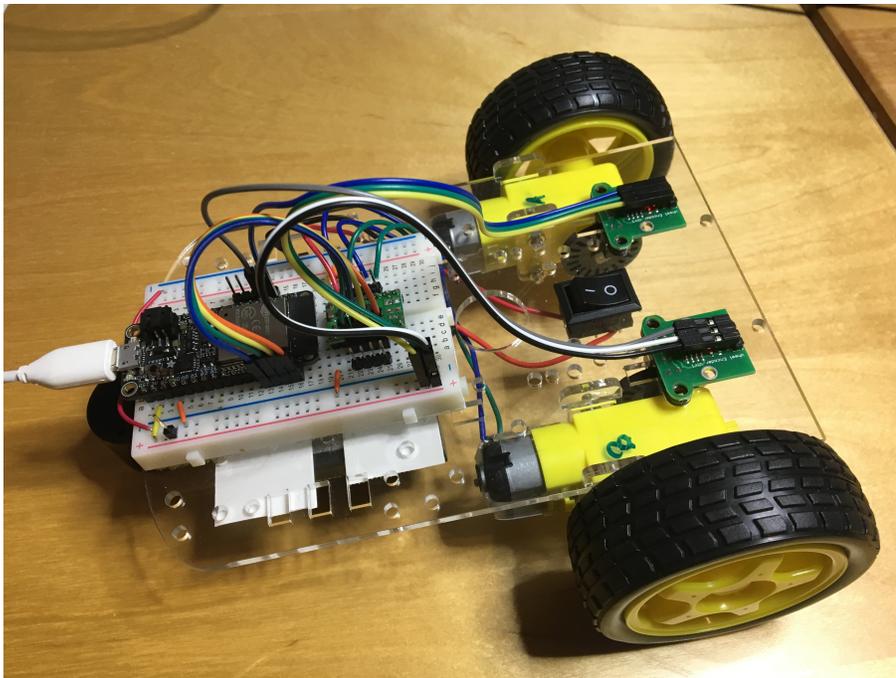


Figure 1 Assembled Robot.

You may either use the stranded wires included in the kit to connect to the motors or solid wires. The former are difficult to insert into the prototyping board while stranded wire is stiff: be extra careful not to apply force on the solder lugs of the motor, they break off easily! Also solder the switch in series with the positive (red) connection of the battery pack.

The battery pack accepts four AA batteries (rechargeable preferred). Check with the DMM that approximately 6 V appear at the battery pack output. If not, verify that the batteries are tight against the connectors of the battery holder. **Be careful not to short the battery.**

Solder Headers

Solder headers to the DRV8833 and MPU9250.

Test Robot

Do not yet insert the AA batteries or set the power switch to off.

Encoders

Use the following code to initialize the encoders (change A6 and A7 to the pins connected to the rotary encoder outputs):

```
encA_pin = Pin(A6)
encB_pin = Pin(A7)

encA = ENC(0, encA_pin)
encB = ENC(1, encB_pin)
encA.filter(1023)
encB.filter(1023)
```

The filter statements are for debouncing.

Then write a loop that prints out the count of `encA` and `encB` and resets the counters every second. Run the program and carefully turn the wheel of one, then the other motor. Watch the red LEDs of the encoder boards blink, and the encoder count to increase. Why does the count keep increasing even when you turn the wheels backwards?

Motors—Voltage Control

Graph motor speed (counts per second, CPS) versus voltage. Vary the motor voltage from 0 . . . 6 V using the lab power supply. Also check the motor current. What are the minimum voltage and corresponding CPS at which the motors start turning?

Checkoff: (20 points)

Motors—PWM Control

Repeat the above experiment, but power the motors from the battery pack and control the speed with PWM. Graph counts per second (from the encoder) versus duty cycle. What are the minimum duty cycle and CPS at which the motors start turning?

Write a loop that slowly varies the motor speed from full speed backwards, through zero, to full speed forward.

Checkoff: (30 points)

IMU

Connect the IMU to the Huzzah32 and run the test program written in the prelab. Verify that the parameters change following orientation and motion.

Checkoff: (20 points)

You now have all the elements—sensing, actuation (motors), control and communication—for building exciting IoT applications!