Final Hack: Team Turret

Andy Wong
Glen Chou
Lydia Lee

Due: May 5, 2015

Contents

1 Features

1.1 Rotating Platform

The user controls the rotation of the platform via joystick. Under the (very figurative) hood, a motor with a protected H-bridge (first analog feature) turns the platform, while two potentiometers serve as the control system (second analog feature) to link the user and platform.

1.2 Rubber Band Gun

Much like a standard rubber band gun, the firing mechanism is a rotating wheel, except ours is controlled by a button that turns a motor. The band is held in a loaded position by a stopped gear. Pressing the button rotates the motor attached to the wheel holding the gear in place and fires the rubber band.

2 Schematics

2.1 Overview

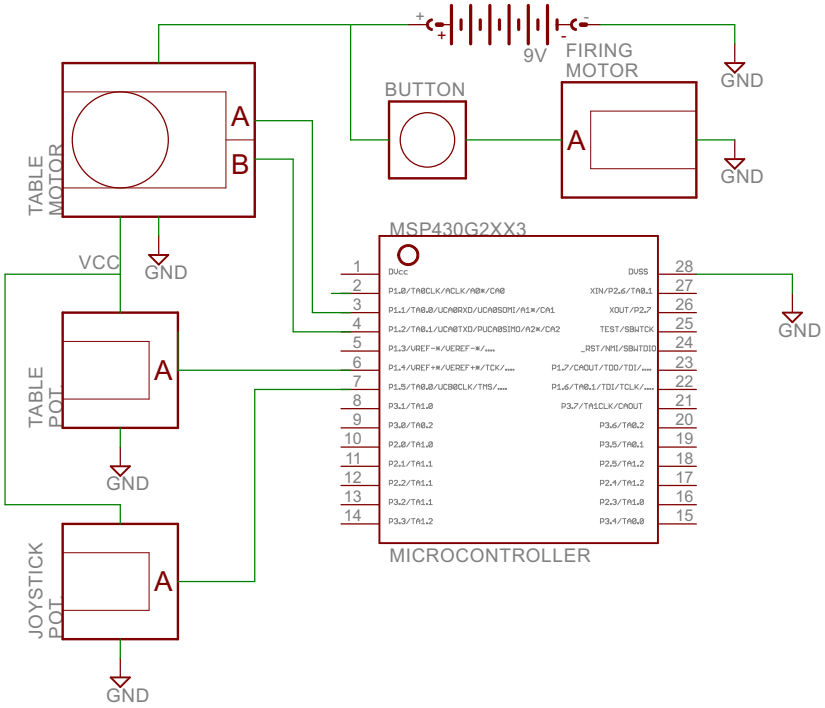

Figure 1: High Level Diagram

2.2 Table Motor

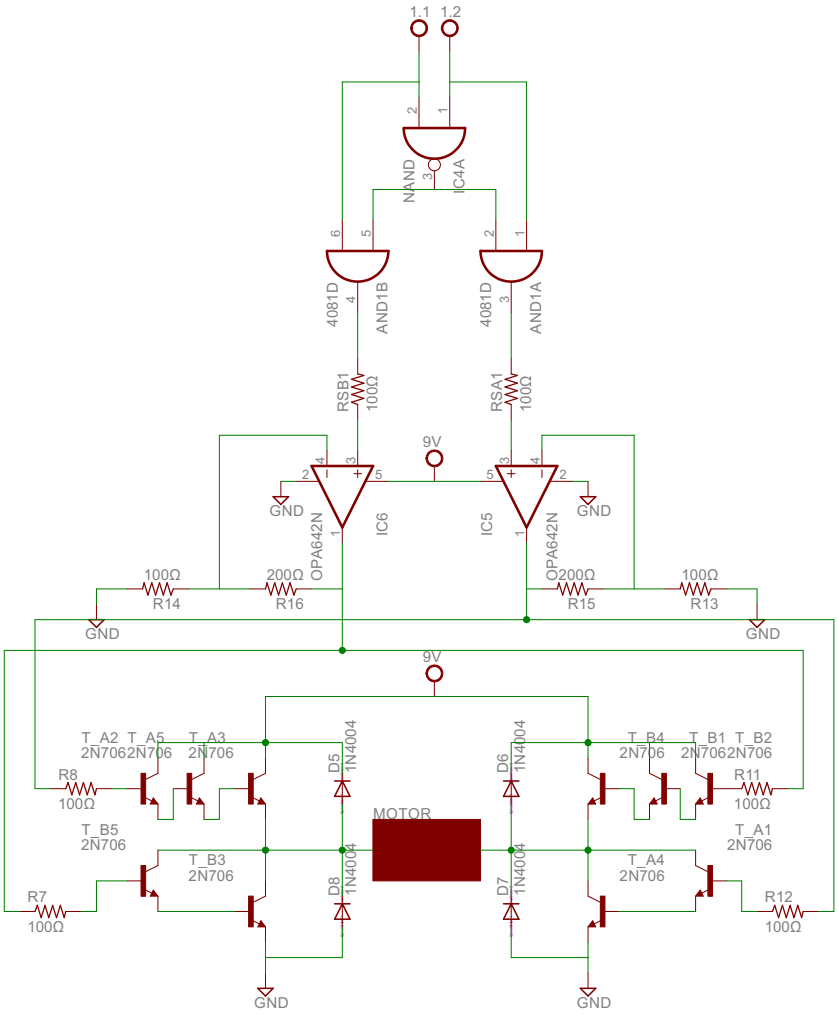

Figure 2: Table Motor Control

The motor control consists of an H-bridge (bottom half) for rotation in both directions. The motor is protected from double-input by logic gates (top half). The initial construction of the H-bridge did not provide sufficient power to drive the motor, so we used more transistors to make Darlington pairs (instead of restarting the bridge with PNP transistors).

2.3 Potentiometers

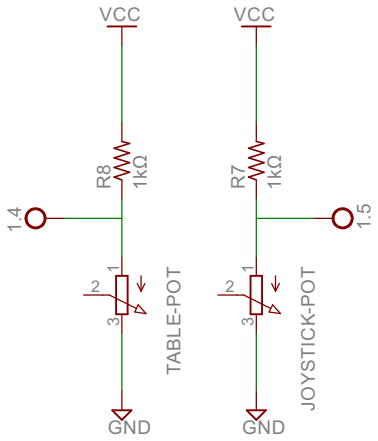

Figure 3: Potentiometers

The MSP analogReads the voltage between the resistor and the potentiometer to determine the rotation of the potentiometer. That in turn is used as input for control to match the two potentiometers.

2.4 Firing Button

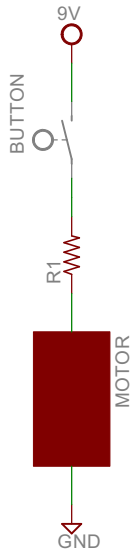

Figure 4: Firing Button

The button and its protection are directly linked to the motor. The motor used for this obviated the need for a diode/transistor protection setup.

3 Code

```
/***************/
/*** code.c ***/
/***************/

// Pins
int motor1a = P1_1; // Don't use 2_3 or 2_4
int motor1b = P1_2;
int potJoy = P1_5;
int potSensor = P1_4;

// Constants
int sensorTolerance = 0.1;

int main(){
  // LEDs used for testing/demo purposes
  pinMode(GREEN_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);

  pinMode(P1_1, OUTPUT); // motor one way
  pinMode(P1_2, OUTPUT); // motor other way
  pinMode(P1_5, INPUT);  // joystick potentiometer
  pinMode(P1_4, INPUT);  // table potentiometer

  int potJoyVal = 0;
  int potSensorVal = 0;

  while(1){
    // Reads the two potentiometer values
    potJoyVal = analogRead(potJoy);
    potSensorVal = analogRead(potSensor);

    // Handles differences in potentiometer readings to choose the
    // direction of rotation
    if(abs(potJoyVal - potSensorVal) < sensorTolerance){
      digitalWrite(motor1a, LOW);
      digitalWrite(motor1b, LOW);
      digitalWrite(RED_LED, LOW);
      digitalWrite(GREEN_LED, LOW);
    }
    else if (potJoyVal > potSensorVal){
      digitalWrite(motor1a, HIGH);
      digitalWrite(motor1b, LOW);
      digitalWrite(RED_LED, HIGH);
      digitalWrite(GREEN_LED, LOW);
    }
    else {
      digitalWrite(motor1a, LOW);
      digitalWrite(motor1b, HIGH);
      digitalWrite(RED_LED, LOW);
      digitalWrite(GREEN_LED, HIGH);
    }
    delay(100);
  }
}
```