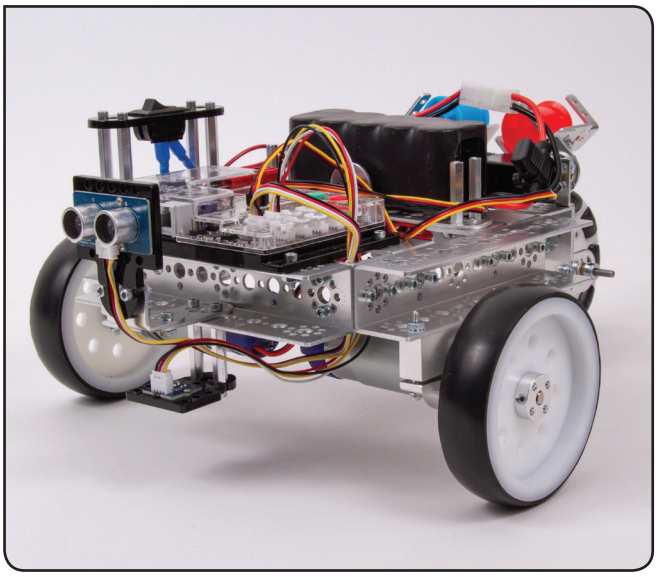
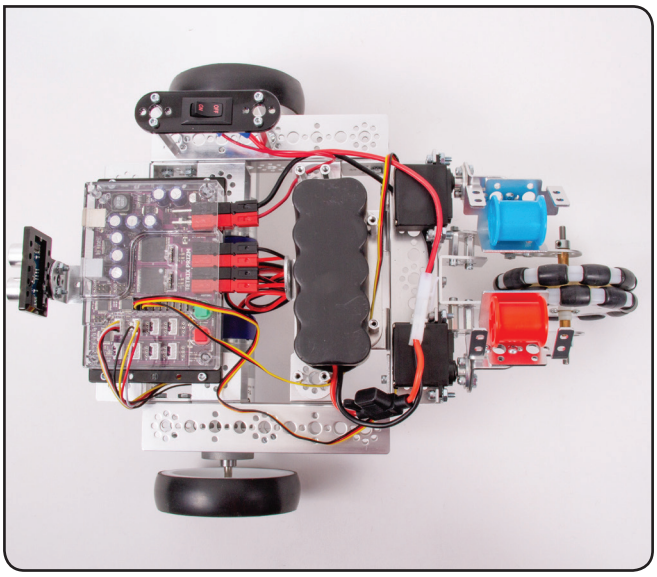
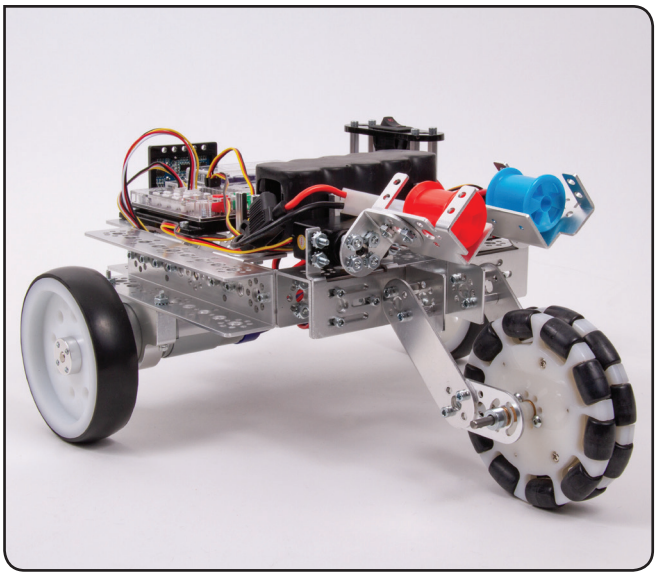
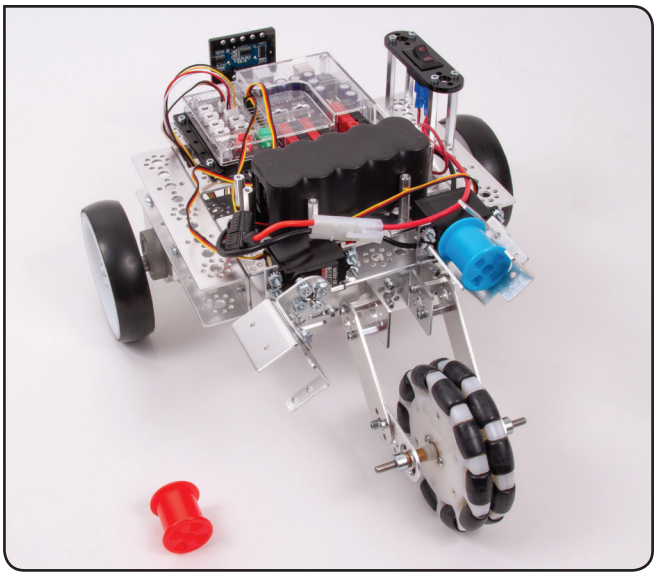
Coding solutions to the Competition 3 challenge will differ greatly depending on the design of the robot, scoring strategies, solution methods, what engineering trade-offs are made, and many other factors. The sample code below works well for the robot design shown here.









Sample Code

```
  #include <PRIZM.h>                                  // include PRIZM library
  PRIZM prizm;                                        // instantiate a PRIZM object "prizm" so we can use its functions

  int normPower = 30;                                 // normal motor power
  int lowPower = 10;                                  // low motor power
  int leftCount = 0;                                  // counts the number of times the robot turns left
  int rightCount = 0;                                 // counts the number of times the robot turns right
  int maxIt = 3;                                      // maximum number of iterations allowed
  int spool = 1;                                      // tracks which servo to activate to dump a spool
  int spoolCarry = 160;                               // servo position to carry spools
  int spoolDump = 0;                                  // servo position to dump spools
  int spoolPos = 0;                                   // storage place to swap positions
void setup() {
  prizm.PrizmBegin();                                   // initialize PRIZM
  prizm.setMotorInvert(1,1);                            // invert the direction of DC Motor 1
  prizm.setServoSpeeds(50,50,50,50,50,50);              // set the speed of all servos
  prizm.setServoPositions(spoolCarry, spoolDump+17,
    spoolCarry, spoolCarry, spoolCarry, spoolCarry);  // set the starting position of all servos for loading
  delay (5000);                                         // wait 5 seconds so spools can be loaded
}
void loop() {

  while(prizm.readSonicSensorCM(2) > 15) {          // loop while no wall detected
    if(prizm.readLineSensor(3) == 0){               // beam reflected, no line detected
      prizm.setMotorPowers(125,normPower);          // turn the robot left
      prizm.setRedLED(LOW);                         // turn off the red LED
      prizm.setGreenLED(LOW);                       // turn off the green LED
      leftCount = leftCount + 1;                    // count number of times in a row no line was detected
      if (leftCount > maxIt) {                      // determine if robot has lost the line
        while (prizm.readLineSensor(3) == 0) {      // repeat while no line detected
          prizm.setMotorPowers(-lowPower,lowPower); // pivot robot sharply to the left
          prizm.setRedLED(HIGH);                    // turn on the red LED
        }
        leftCount = 0;                              // reset the leftCount variable
        rightCount = 0;                             // reset the rightCount variable
      }
    }
    if(prizm.readLineSensor(3) == 1){               // no beam reflected, line detected
      prizm.setMotorPowers(normPower,125);          // turn the robot right
      prizm.setRedLED(LOW);                         // turn off the red LED
      prizm.setGreenLED(LOW);                       // turn off the green LED
      rightCount = rightCount + 1;                  // count number of times in a row the line was detected
      if (rightCount > maxIt) {                     // determine if robot is in the middle of the line
        while (prizm.readLineSensor(3) == 1) {      // repeat while the line is detected
          prizm.setMotorPowers(lowPower,-lowPower); // pivot robot sharply to the right
          prizm.setGreenLED(HIGH);                  // turn on the green LED
        }
        rightCount = 0;                             // reset the rightCount variable
        leftCount = 0;                              // reset the leftCount variable
      }
    }
  }
  aboutTurn();                                      // survivor has been detected, run the aboutTurn function
  delay (1000);                                     // wait 1 second
  dumpSpool();                                      // run the dumpSpool function to deliver spool to survivor
  delay (1000);
}                                                   // end the main loop
void aboutTurn(){                                       // this function turns the robot around when a survivor is detected
  prizm.setMotorPowers(125,125);                        // stop the motors
  delay(500);
  prizm.setMotorPowers(lowPower,-lowPower);             // pivot robot sharply to the right
  delay(1000);
  while (prizm.readLineSensor(3) == 0){                 // repeat until the line is detected
    prizm.setMotorPowers(lowPower,-lowPower);           // turn until the line is detected
  }
  delay(300);
  prizm.setMotorPowers(125,125);                        // stop the motors
}
void dumpSpool(){                                       // this function dumps a spool each time it is run
  prizm.setServoPosition(spool, spoolDump);             // dump the designated spool
  delay (3000);
  prizm.setServoPositions(spoolCarry, spoolDump+17,     // reset the position of all servos to carry position
    spoolCarry, spoolCarry, spoolCarry, spoolCarry);
  spool = spool + 1;                                    // increase spool count by 1 to dump next spool on next iteration
  spoolDump = spoolCarry;                               // change dump and carry positions for the next spool because...
  spoolCarry = spoolPos;                                // ...the two dump servos are reversed
}
```