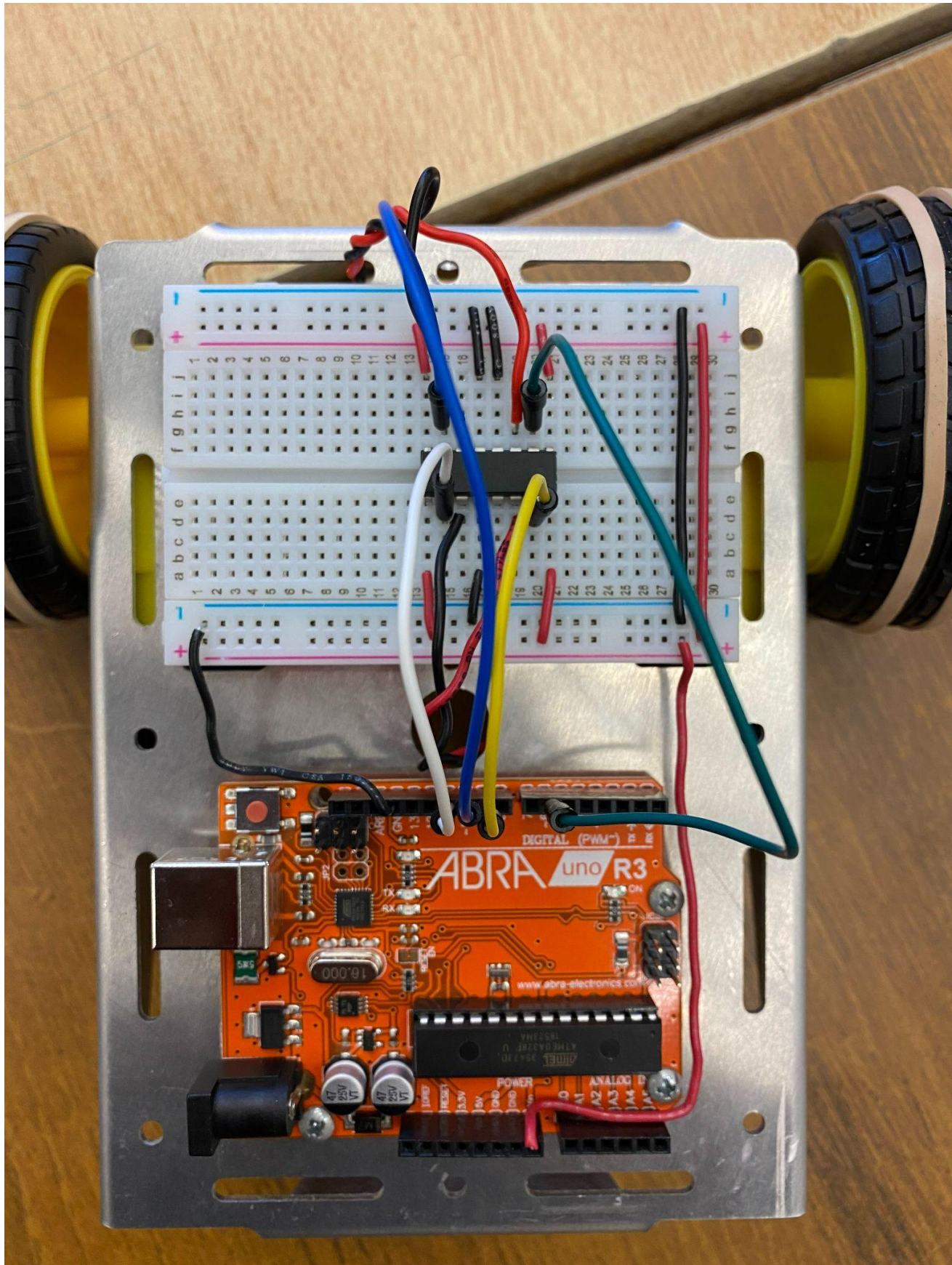


LVL	Criteria
R	
1	
2	
3	
4	<p>"build and wire"[3]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> circuit is correct, routed cleanly and easy to follow[1½]</li> <li><input type="checkbox"/> all full voltage wire red and all gnd wires black</li> <li><input type="checkbox"/> signal wire colours chosen to allow easier tracing of circuit[½]</li> </ul> <p>tinkerCAD[2]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> all components mounted on breadboard and do not block view of other components[½]</li> <li><input type="checkbox"/> wires horizontal or vertical only with 90 degree bends[½]</li> <li><input type="checkbox"/> wires do not cross in front or behind other components or component terminals and do not run on top of one another[½]</li> <li><input type="checkbox"/> wires and component do not share the same hole on the breadboard and wires do not cross when possible[½]</li> </ul> <p>in person[2]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> all full voltage and gnd wires are solid core, flat to breadboard, horizontal or vertical with 90 degree bends</li> <li><input type="checkbox"/> solid core wires stripped 6-8mm[½]</li> <li><input type="checkbox"/> no bare wire visible [½]</li> </ul> <p>"programming"[3]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> final "test your understanding" complete and working correctly[1½]</li> <li><input type="checkbox"/> code commenting is accurate and complete (including title)[½]</li> <li><input type="checkbox"/> program structure and spacing is logical and demonstrates organization[½]</li> <li><input type="checkbox"/> code text submission is courier new font and is coloured to allow easier identification of comments[½]</li> </ul> <p>"inspection questions"[1]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> demonstrates full understanding of circuit and interfacing concepts in conversation with teacher</li> </ul>
4+	<p>"enhancements"[1]</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> minimized number and length of wires and wire crossings[½]</li> <li><input type="checkbox"/> circuit enhancement complete and working correctly[½]</li> </ul>

Build image:



code:

```
/*
Names: Siddarth & Mostafa
Dates: May 11, 2022
Description: Code for interfacing lab 12 - Motors
*/

const int motors[2][2] = {{6,10}, {9,11}}; setting constant 2d array for motor pins
//function for straight movement (also controls if the motors stop)
void straight(bool forward = true, int time = 3000, bool off = false) {
    int forVar = forward ? 255 : 0; //one line if statement that controls direction;
    //inline one lined if statements used to halt vehicle if off is true
    analogWrite(motor[0][0], (off ? 0 : forVar)); // right motor forward
    analogWrite(motor[0][1], (off ? 0 : 255-forVar)); // right motor back
    analogWrite(motor[1][0], (off ? 0 : forVar)); // left motor forward
    analogWrite(motor[1][1], (off ? 0 : 255-forVar)); // left motor back
    delay(time); // delay (usually 3 seconds by default)
}

//function for side movement
void side(bool forward = true, bool right = true, int time = 3000) {
    int leftVar = right ? 75 : 255; //sets the speed for left motor
    int rightVar = right ? 255 : 75; //sets the speed for right motor
    int forVar = forward ? 1 : 0; //controls direction (sets which motor pin is on)

    analogWrite(motor[0][1-forVar], rightVar); //the side that is on for right motor
    analogWrite(motor[0][forVar], 0); //the side that is off for right motor
    analogWrite(motor[1][1-forVar], leftVar); //the side that is on for left motor
    analogWrite(motor[1][forVar], 0); //the side that is off for left motor
    delay(time); //delay
}

void setup() { //setup
    pinMode(motor[0][0], OUTPUT); //sets right motor first pin as output
    pinMode(motor[0][1], OUTPUT); //sets right motor second pin as output
    pinMode(motor[1][0], OUTPUT); //sets left motor first pin as output
    pinMode(motor[1][1], OUTPUT); //sets left motor second pin as output

    straight(); // forward
    straight(false); // backwards
}
```

```
side(); //forward + right
side(false); //backward + right
side(false, false); // backward + left
side(true, false); // forward + left
straight(true, 1, true); //stop
}

void loop() {} //loop (empty, thus does nothing as it loops)
```