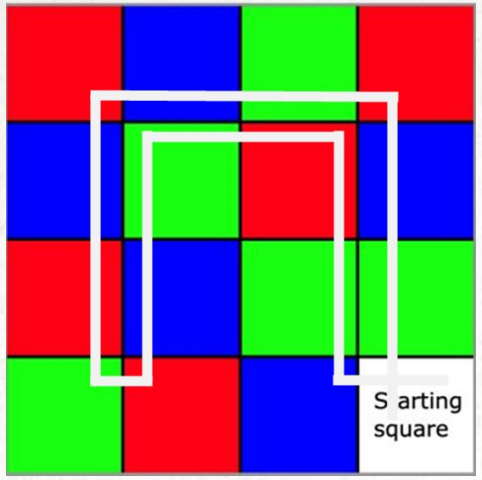
CURRENT STATUS ---------- FUTURE WORK

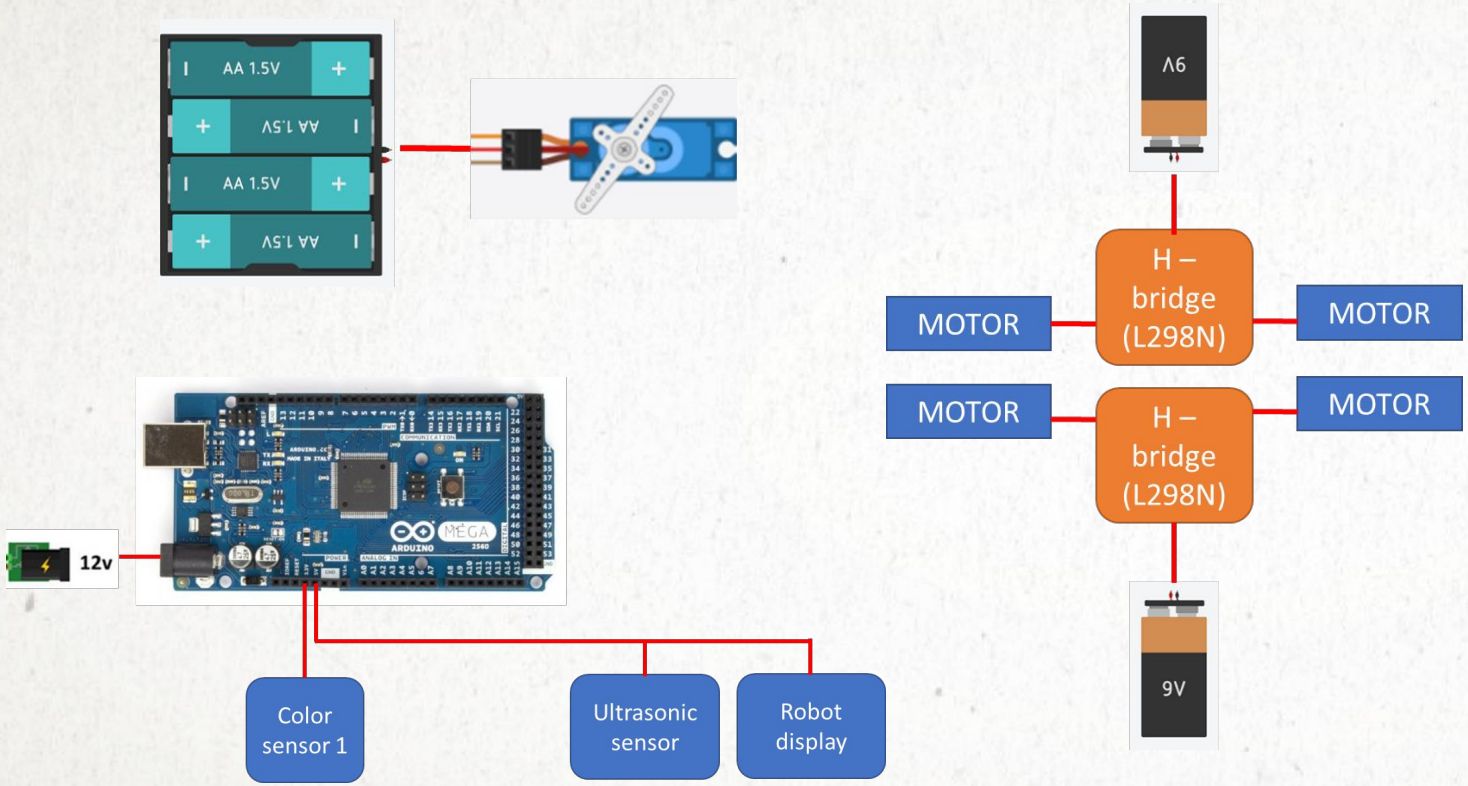
- Made final decision on how the robot will complete the Mole Whacker task: Go through each square



- Ordered necessary components for project
- Began creating prototype for the Mole Whacker placing mechanism
- Completed initial circuit diagram
- Completed initial FSM with Arduino code

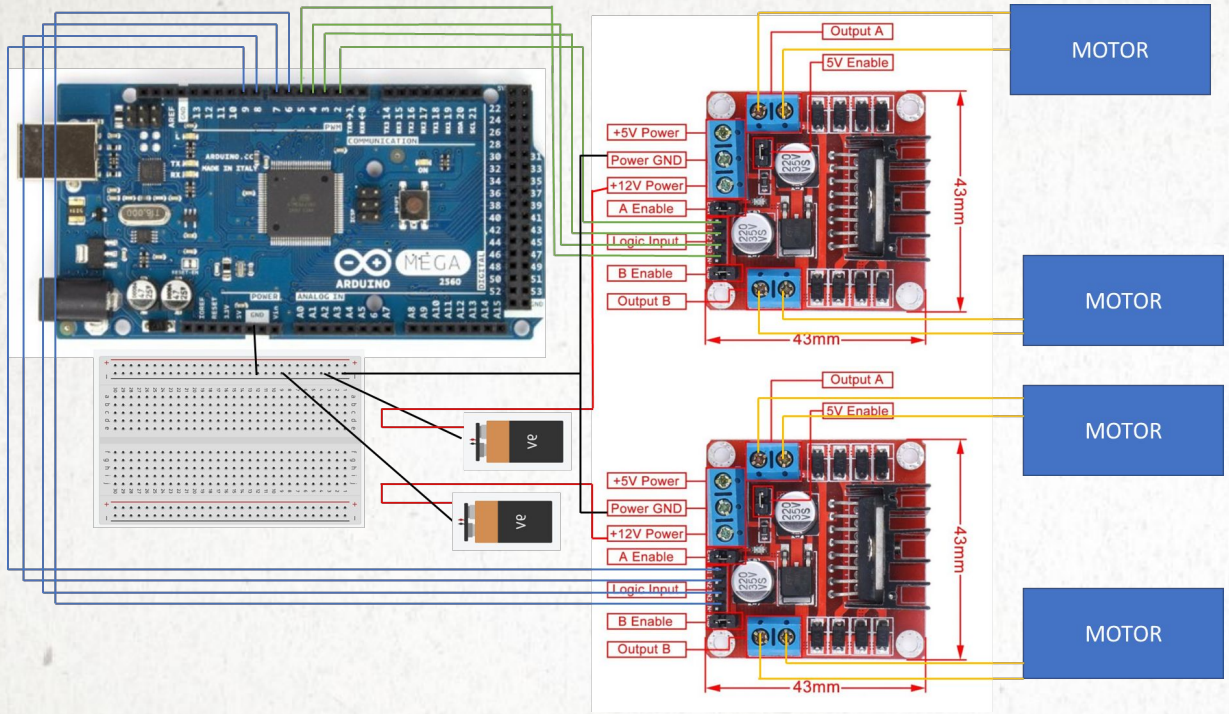
- **Week 11:** Begin prototyping and testing subsystems
- **Week 12:** Continue testing subsystems and integrating system
- **Week 13:** Testing of whole system and debugging
- **Week 14:** More testing and debugging
- **Week 15:** Final modifications

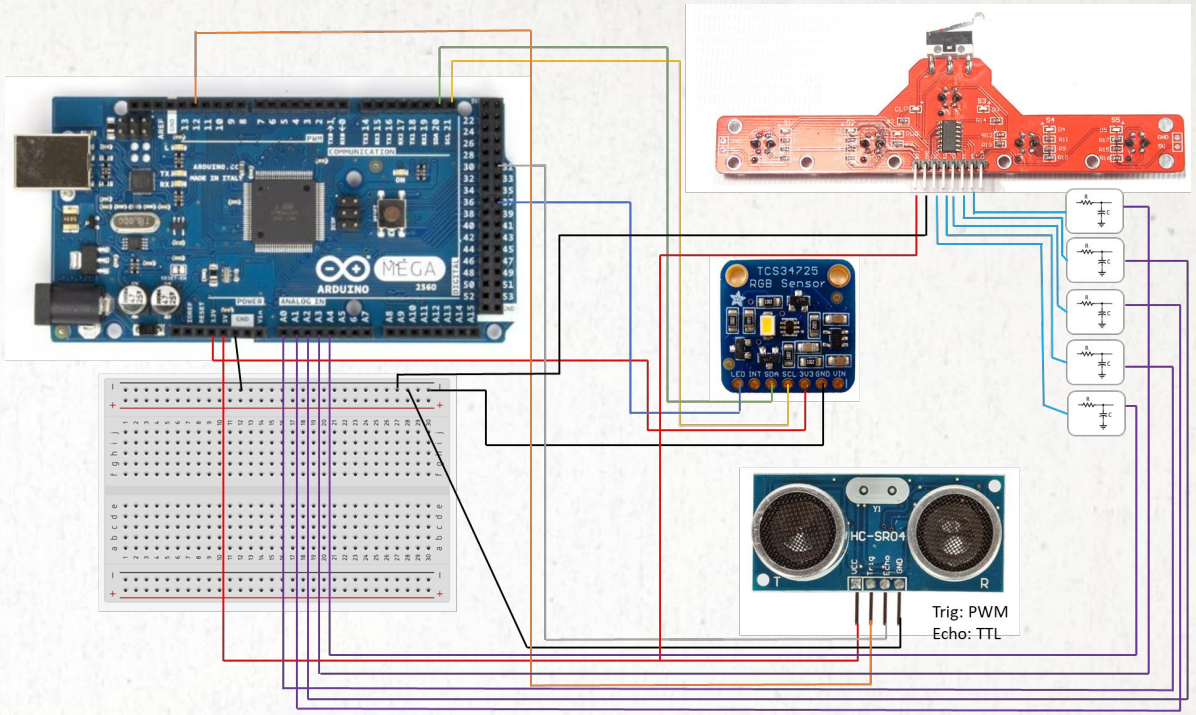
CIRCUIT DIAGRAM – POWER DISTRIBUTION



- 12 V lithium battery powers Arduino mega
- Arduino mega powers sensors and robot display(LEDs)
- Two 9V batteries power H-bridge and motor
- Switches attached to 9V battery packs act as emergency stop to turn motors off

CIRCUIT DIAGRAM – DRIVE MECHANISM & SENSOR

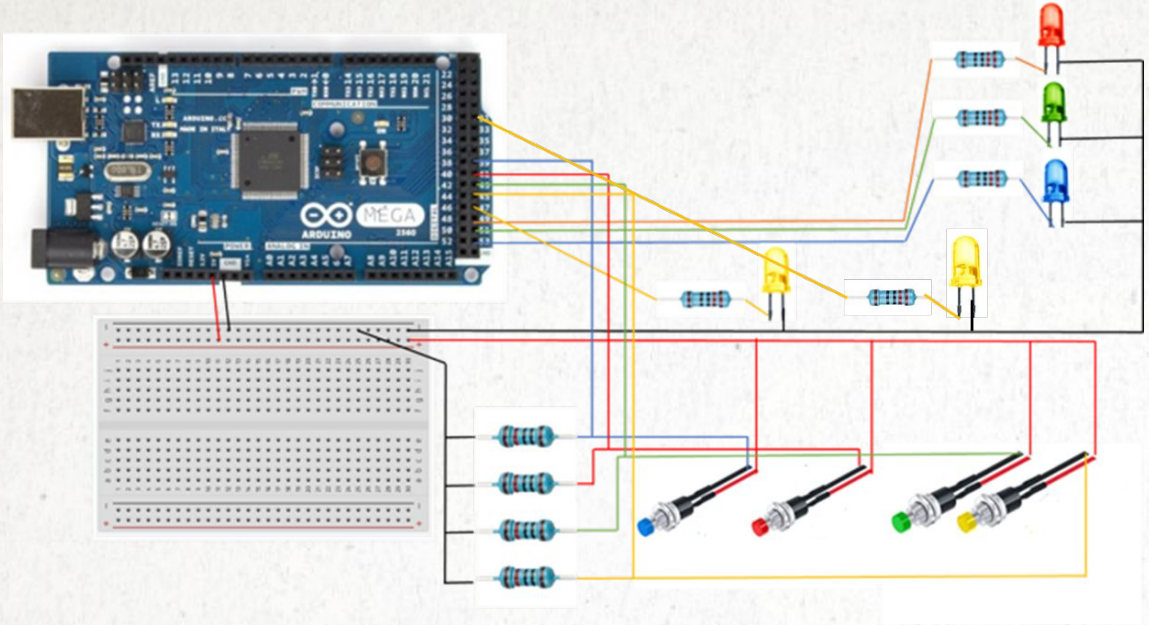




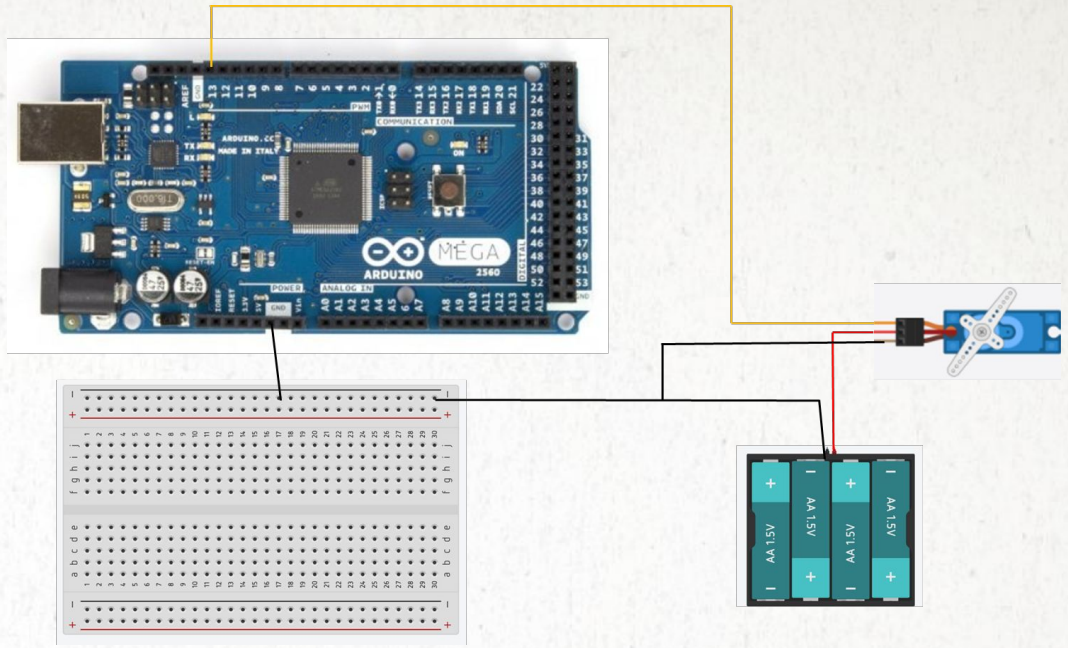
- Two L298N H-bridges and 4 motors
- Arduino provides logical inputs for H-bridges

- One color sensor, one line follower sensor and, one ultrasonic sensor
- RC low pass filters filter the outputs from line follower sensor

CIRCUIT DIAGRAM – ROBOT DISPLAY & MOLE DISTRIBUTION

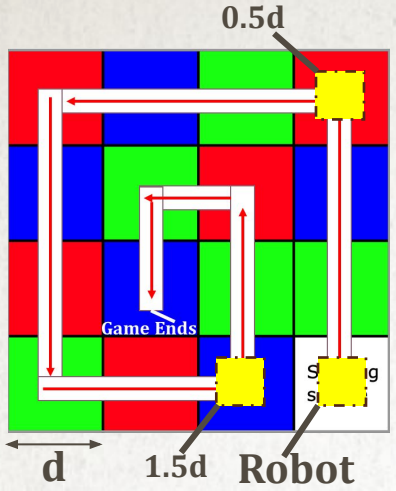


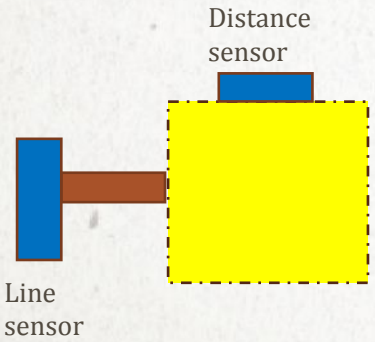
- LED indicating if it is started
- LEDs indicating target color
- LED indicating if the current color matches the target color.
- Color buttons - Send target color depending on button pressed
- Start button – enter game mode



- Arduino PWM output control servo motor for mole distribution mechanism

FSM: INPUTS AND OUTPUTS

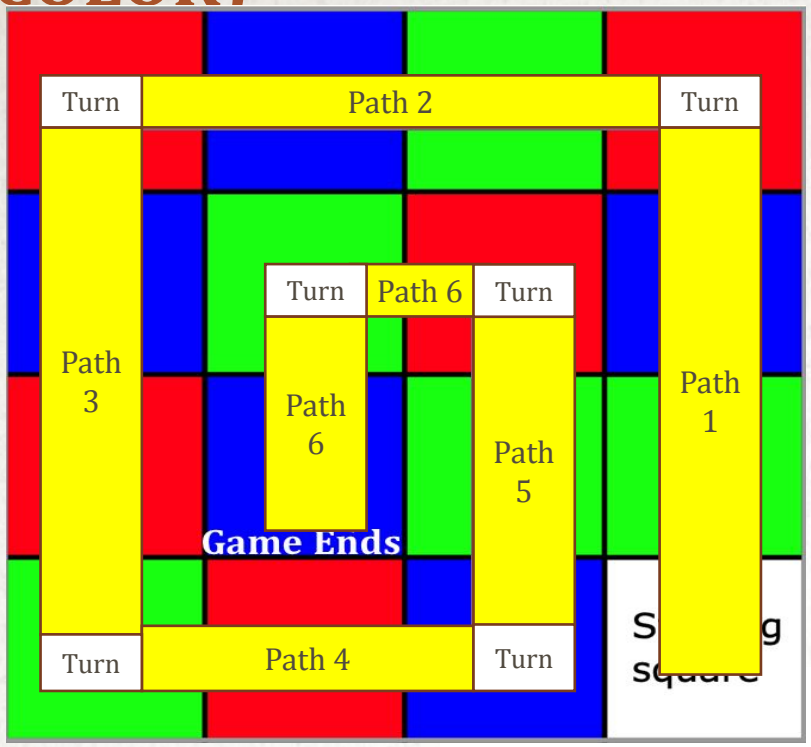




Red, Green, Blue Button	**Target Color** 0 and 1
Emergency Stop	**Stop if Pressed** 0 and 1
Start Button	**Start the Game** 0 and 1
Color Sensor	**Detects Respective Colors** 100 (R), 010 (G), 001 (B)
Distance Sensor	**Four Different States** 00 (=0.5d), 01 (>0.5d), 11 (1.5d), 10 (>1.5d)
Line Sensor	**Binary Line Sensor (Assumption)** 1 (Line), 0 (No Line)
New Square Sensor	**Movement to New Square** 1 (Yes), 0 (No)

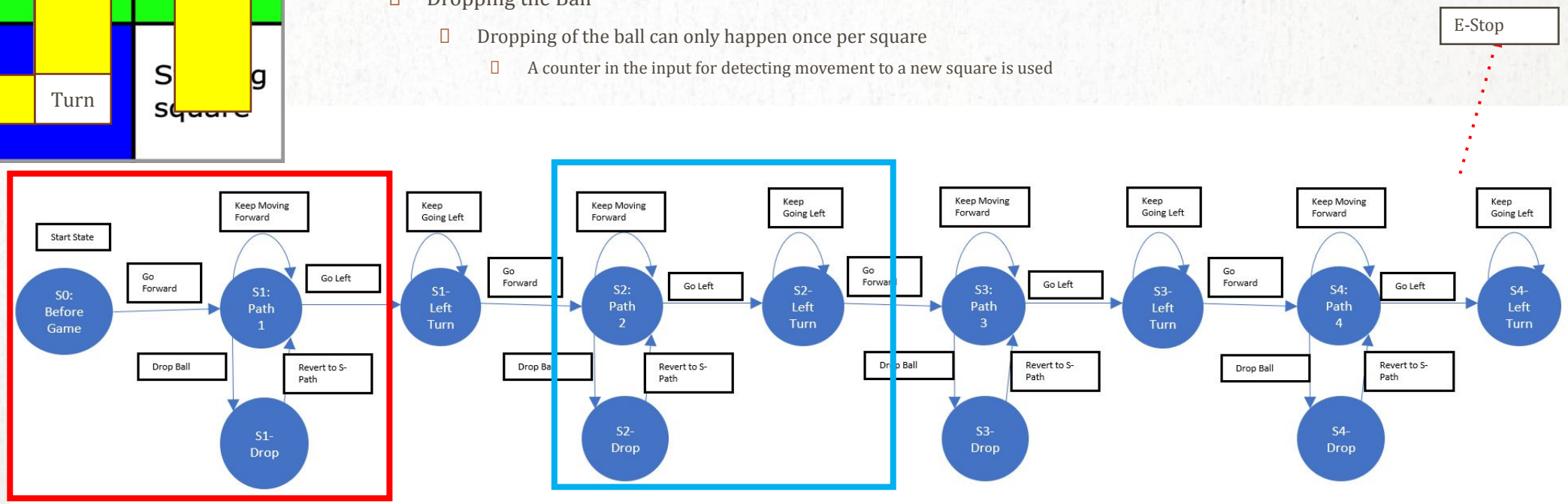
Actuator	**Drop/Not Drop Mole** 0 (OFF) and 1 (ON)
Motor	**Motor State** 0 (OFF) and 1 (ON)
Wheel Drive	**Movement Modes** 0 (Straight) and 1 (Left)
LEDs (3)	**Target Color Indicator** 0 (OFF) and 1 (ON)
LED (1)	**Match Indicator** 0 (No) and 1 (Yes)
LED (1)	**Game Mode** 0 (No) and 1 (In Game)

FSM (EXAMPLE: DETECTING RED COLOR)

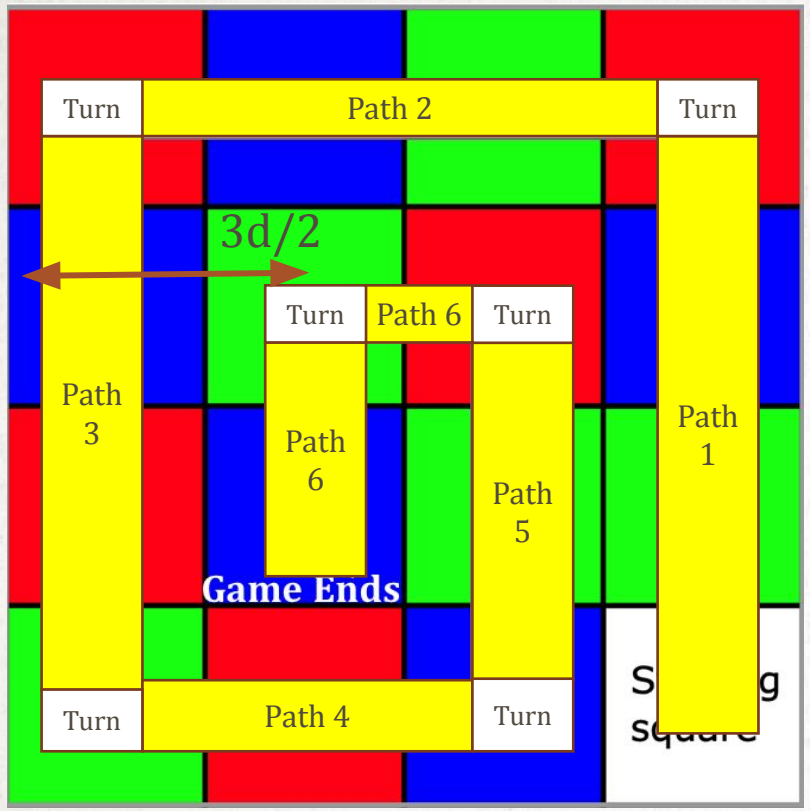


d/2

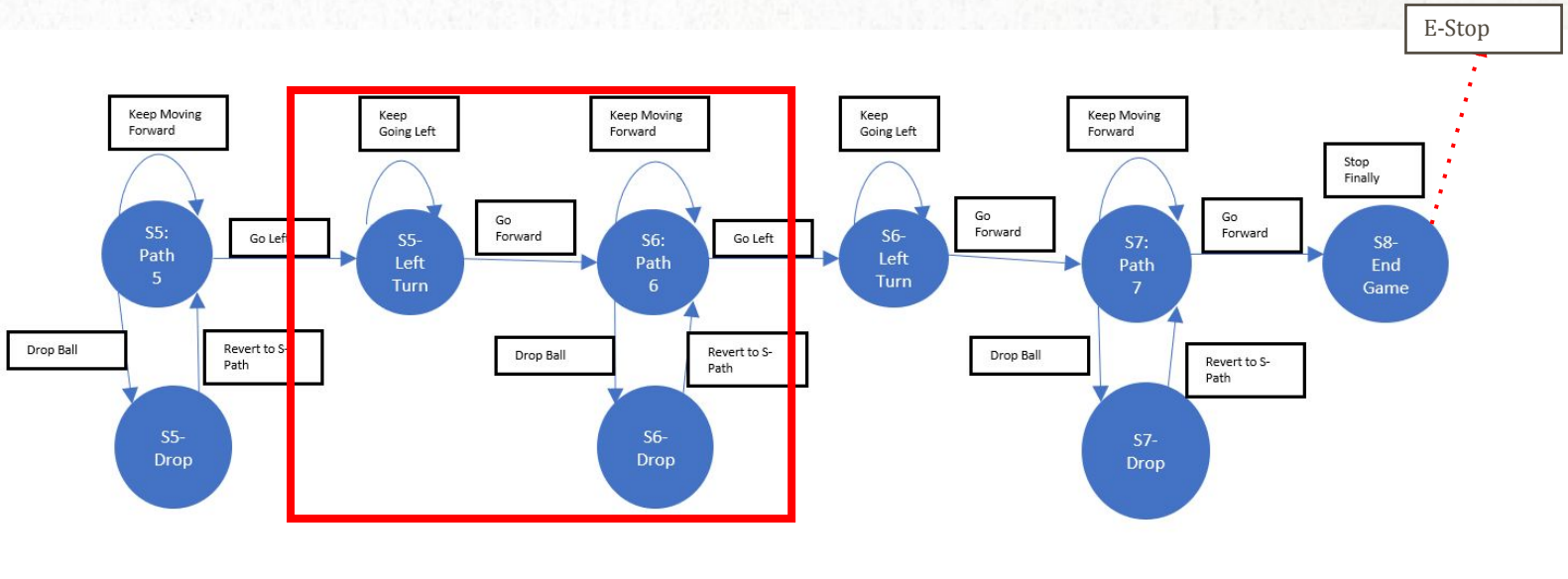
- Every Square Method
- Every state connected to a stop state represented by dotted line
  - From **S0 to S1: Path 1**
    - The robot will move forward while also ensuring following the line to the left of the robot
    - The robot will keep moving forward till it reaches a distance of **d/2** away from the outer boundary
  - Turning Method: Example in **S2: Path 2 to S2-Left Turn**
    - The initial turn takes the robot away from the line to its left
    - It then turns left till it finds the new line
  - Dropping the Ball
    - Dropping of the ball can only happen once per square
      - A counter in the input for detecting movement to a new square is used



FSM (EXAMPLE: DETECTING RED COLOR)



- Every Square Method
- From **S5: Left Turn** to **S6: Path 6**
  - The robot will move forward while also ensuring following the line to the left of the robot
  - The robot will keep moving forward till it reaches a distance of **3d/2** away from the outer boundary
- Modifications
  - The logic for turning is dependent on the sensing of the line to its left
    - Robot may not turn and instead keep moving forward is a possibilities since we depend on the line
  - A ball could be dropped twice if not careful about the counter for the new square when turning



FLOW OF CODE

Assigning Input Pins

```
//Input Pins
int Pin1 = 2;  int Input_1 = 0;
int Pin2 = 3;  int Input_2 = 0;
int Pin3 = 4;  int Input_3 = 0;
int Pin4 = 5;  int Input_4 = 0;
int Pin5 = 6;  int Input_5 = 0;
int Pin6 = 7;  int Input_6 = 0;
int Pin7 = 8;  int Input_7 = 0;
int Pin8 = 9;
```

Assigning Output Pins

```
//OutputPins
int Output_1 = 10;
int Output_2 = 11;
int Output_3 = 12;
int Output_4 = 13;
int Output_5 = 14;
int Output_6 = 15;
```

Assigning Pin Modes

```
void setup()

{
//Assigning Pins as Output
  pinMode(Output_1, OUTPUT);
  pinMode(Output_2, OUTPUT);
  pinMode(Output_3, OUTPUT);
  pinMode(Output_4, OUTPUT);
  pinMode(Output_5, OUTPUT);
  pinMode(Output_6, OUTPUT);
```

```
//Assigning Pins as Input
  pinMode(Pin1, INPUT);
  pinMode(Pin2, INPUT);
  pinMode(Pin3, INPUT);
  pinMode(Pin4, INPUT);
  pinMode(Pin5, INPUT);
  pinMode(Pin6, INPUT);
  pinMode(Pin7, INPUT);
  pinMode(Pin8, INPUT);
}
```

Enter the Switch Loop

```
 switch (state)
  {

//////////////////////////////////////////////////////
case STATE_0:

    if (Input_1 == 1 && Input_2 == 0 && Input_3 == 1 && Input_4 == 0 && Input_5 == 0 && Input_6 == 1 && Input_7 == 1) {

      state = STATE_1;
      Serial.println("STATE_1 Moving Forward");
      digitalWrite(Output_1, LOW);
      digitalWrite(Output_2, HIGH);
      digitalWrite(Output_3, HIGH);
      digitalWrite(Output_4,HIGH);
      digitalWrite(Output_5, LOW);
      digitalWrite(Output_6, HIGH);
    }
 break;
```

Recognize Internal Functions

```
case STATE_1:

    if (Input_1 == 1 && Input_2 == 0 && Input_3 == 1 && Input_4 == 0 && Input_5 == 0 && Input_6 == 1 && Input_7 == 1) {

      state = STATE_1;
      Serial.println("STATE_1 Moving Forward");
      digitalWrite(Output_1, LOW);
      digitalWrite(Output_2, HIGH);
      digitalWrite(Output_3, HIGH); // This would call back to a loop which would include the codes for the Motor Drive
      digitalWrite(Output_4,HIGH);
      digitalWrite(Output_5, LOW);
      digitalWrite(Output_6, HIGH);
    }
```

Separate Function to be Written

Make State Transitions

```
case STATE_1:

    if (Input_1 == 1 && Input_2 == 0 && Input_3 == 1 && Input_4 == 0 && Input_5 == 0 && Input_6 == 1 && Input_7 == 1) {

      state = STATE_1;
      Serial.println("STATE_1 Moving Forward");
      digitalWrite(Output_1, LOW);
      digitalWrite(Output_2, HIGH);
      digitalWrite(Output_3, HIGH); // This would call back to a loop which would include the codes for the Motor Drive
      digitalWrite(Output_4,HIGH);
      digitalWrite(Output_5, LOW);
      digitalWrite(Output_6, HIGH);
    }
```

```
    if (Input_1 == 1 && Input_2 == 0 && Input_3 == 1 && Input_4 == 1 && Input_7 == 1 && Input_8 == 1 ) {

      state = STATE_2;
      Serial.println("STATE_2 Moving Forward");
      digitalWrite(Output_1, HIGH);
      digitalWrite(Output_2, LOW);
      digitalWrite(Output_3, HIGH);
      digitalWrite(Output_4,HIGH);
      digitalWrite(Output_5, HIGH);
      digitalWrite(Output_6, HIGH);
    }
```

CAD AND INITIAL PROTOTYPING