The learners will be able to:
- Identify functions of VEX IQ sensors;
- Explain the utility of sensor data within robotics programming;
- Investigate parameters of using waitUntil command block thru sensor data;
- Apply Boolean logic in programming sensors;
- Program the sensors to limit/affect/cause the robot’s movements;
- Approximate threshold values in programming sensors;
- Evaluate threshold values that make sense within the robot’s scale and environment;
- Assess methods to overcome the effects of sensor delays and momentum;
- Consider methods to overcome low contrast in the sensor’s detection;
- Give and describe instances that uses conditional statement;
- Create programming sequence using sensors thru pseudocodes or flowcharts; and
- Program the physical robot to use different sensors

### VEX IQ AND ITS SENSORS

<table>
<thead>
<tr>
<th>SENSOR</th>
<th>FUNCTION</th>
<th>PORT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bumper Switch</td>
<td>Gives the Robot Brain a signal when they are pressed, such as from a robot hitting a wall or an arm mechanism hitting its end.</td>
<td>8</td>
</tr>
<tr>
<td>Touch Led</td>
<td>Glows in millions of colors, and respond when the top surface is touched.</td>
<td>2</td>
</tr>
<tr>
<td>Distance Sensor</td>
<td>Uses ultrasonic (above audible human hearing range) sound waves to detect its range to objects.</td>
<td>7</td>
</tr>
<tr>
<td>Color Sensor</td>
<td>Detects the color hue and greyscale value of objects in close proximity.</td>
<td>3</td>
</tr>
<tr>
<td>Gyro Sensor</td>
<td>Measures the turn rate and calculates the current heading. This makes it easier to perform specific tasks like a robot turning exactly 90 degrees</td>
<td>4</td>
</tr>
</tbody>
</table>
A. Distance Sensor
✓ Also known as ULTRASONIC SENSOR
✓ Detects obstacles/objects at a specific distance
✓ Connected to Port 7 (Clawbot IQ)
✓ The values used are called THRESHOLD VALUE
✓ Threshold Value: 1 to 600 mm

DISTANCE SENSOR: Command Blocks

**setMultiplemotors**

```haskell
setMultipleMotors([-50, leftMotor], rightMotor, noMotorOnPort, noMotorOnPort);
```

- Turn on the motor 1 (left motor) and motor 6 (right motor)
- Since the ultrasonic sensor is located at the back of the robot, the speed is set to NEGATIVE POWER LEVEL (to do a backward movement)

**waitUntil**

```haskell
waitUntil (Select a Value == 0);
```

- first parameter set it into getDistanceValue(distanceMM)

- second parameter to create a move until near program, the LESS THAN OR EQUAL TO symbol
- third parameter allows you to specify the threshold value. Threshold value depends on from what distance you want the ultrasonic sensor detects the wall/object. This sensor reads distance in millimeters (mm).

**stopMultiplemotors**

```haskell
stopMultipleMotors(rightMotor, leftMotor, , , , );
```

- Turn off the left and right motor

**Move Until Near**

- Programming the robot to move BACKWARD until the robot is in a specific distance from the wall/object
- Use the LESS THAN OR EQUAL TO symbol.
Move Until Far

- Using the same command blocks, the robot needs to move FORWARD from the object.
- Use the GREATER THAN OR EQUAL TO symbol.

B. Gyro Sensor

- Measures rotational angle of the robot.
  - Assuming that the sensor is mounted on the chassis of the robot.
  - The sensor measures rotation along the axis indicated by the curved arrow.
- Returns values in degrees.
  - **Counter-clockwise movement increases the value, clockwise movement decreases.**
- Is a MEMS Sensor (Micro-Electro-Mechanical System).
- **Must** be reset at the beginning of the program.
- Default Port: Port #4

    ```
    1. resetGyro (gyroSensor);
    2. wait (1, seconds);
    3. setMotor (motor1, -50);
    4. setMotor (motor6, 50);
    5. waitUntil (getGyroDegrees(gyroSensor) >= 90);
    6. stopMultipleMotors (motor1, noMotor, noMotor, noMotor, noMotor); 
    ```

C. Color Sensor

- Has 5 internal sensors that allow it to detect different wavelengths of light and return:
  - Hue Values / Simple Color Names
  - Grayscale Values
  - Proximity Values
- Useful for color detection and line following.
- Is affected by ambient light in the environment, as well as distance to surface/object.

    **Color Names**
    - 12 "simple" color names can be used with the Color Sensor.
      - These color names are the same as the Touch LED's color names.
      - Colors correspond to a range of hue value.

Forward until Color
WORKSHEET # 5
SENSING

Draw 😊 if the program is correct and fix the program and encircle the part making it incorrect.

A. The robot will move forward until the bumper switch releases, then stops.

```plaintext
setMultipleMotors (-50, leftMotor, rightMotor, noMotor, noMotor);
waitUntil (getBumperValue(bumpSwitch) 0);
stopAllMotors();
```

B. The robot will turn to the right.

```plaintext
resetGyro (gyroSensor);
wait (2, seconds);
setMotor (leftMotor, 75);
setMotor (rightMotor, 75);
waitUntil (getDistanceValue(distanceMM) 300);
stopAllMotors();
```

C. The robot will move backward until near 300mm from the wall, then the robot will stop.

```plaintext
setMultipleMotors (-50, leftMotor, rightMotor, noMotor, noMotor);
waitUntil (getDistanceValue(distanceMM) 300);
stopAllMotors();
```

D. The robot will move forward until the color sensor detect color red, then stops and continue moving forward once the light is green.

```plaintext
setMultipleMotors (50, leftMotor, rightMotor, noMotor, noMotor);
waitUntil (getColorName(colorDetector) colorRed);
stopAllMotors();
waitUntil (getColorName(colorDetector) colorRed);
forward (1, rotations, 50);
```