

# Team 6 Project: SHARP-IR Sensor

By Rishi Zaveri

# IR Sensor GP2Y0A02YK0F

This project uses a precision IR sensor to measure distance from 20cm to 150cm. It requires a supply voltage around 5 volts and a ground pin connection. It outputs an analog voltage between 0 and 2.85 volts.

**Distance Measuring Sensor Unit**  
**Measuring distance: 20 to 150 cm**  
**Analog output type**

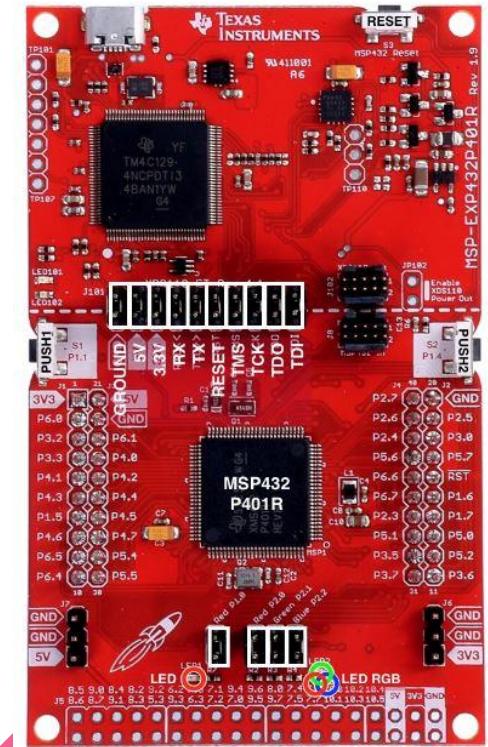


# ADC Converter TI MSP432

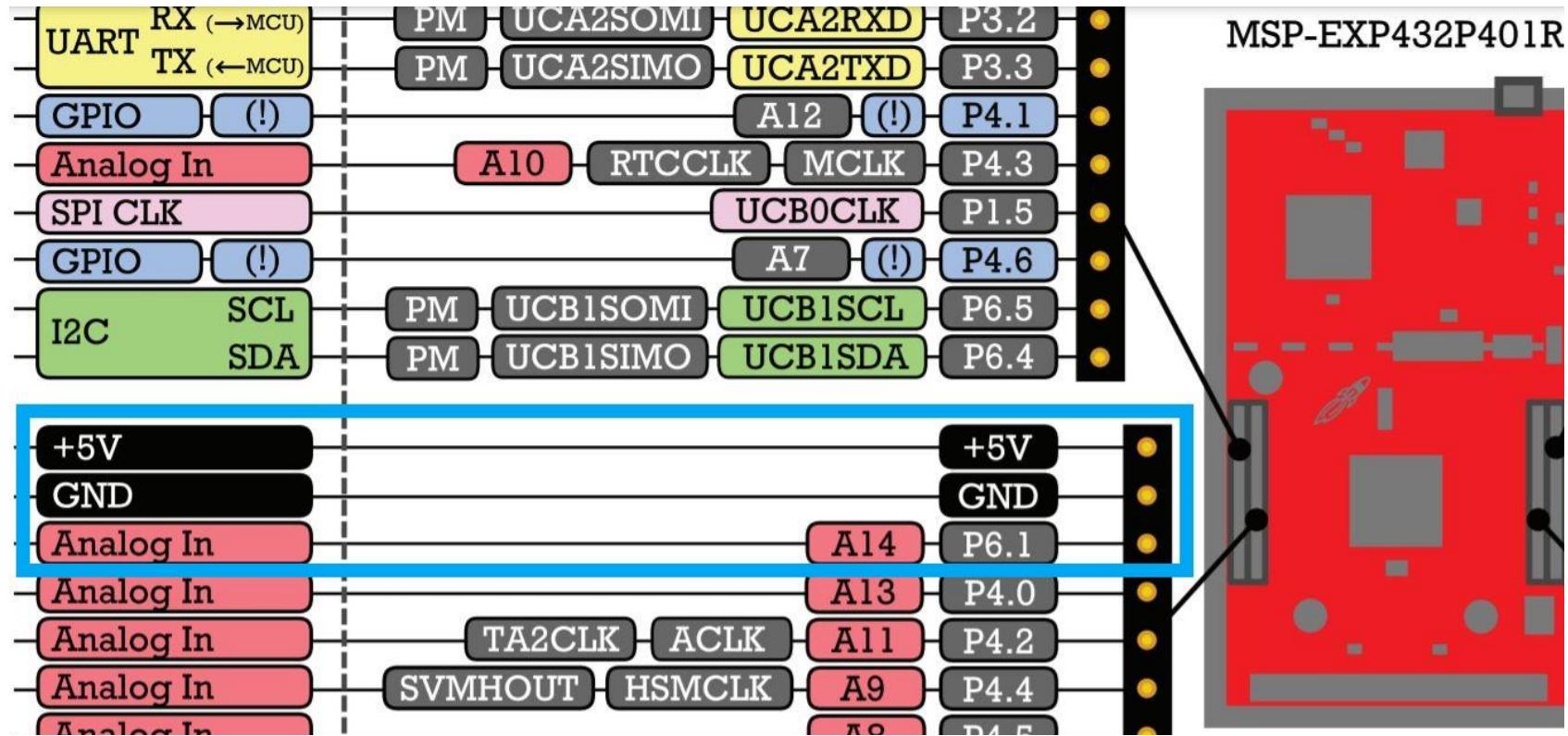
An ADC converter is used to obtain the analog voltage data from the sensor. The USB cable is used for both powering the device and transferring data to the Raspberry Pi. The Energia IDE helps to code this device.

LaunchPad with MSP432P401R

Revision 2.0

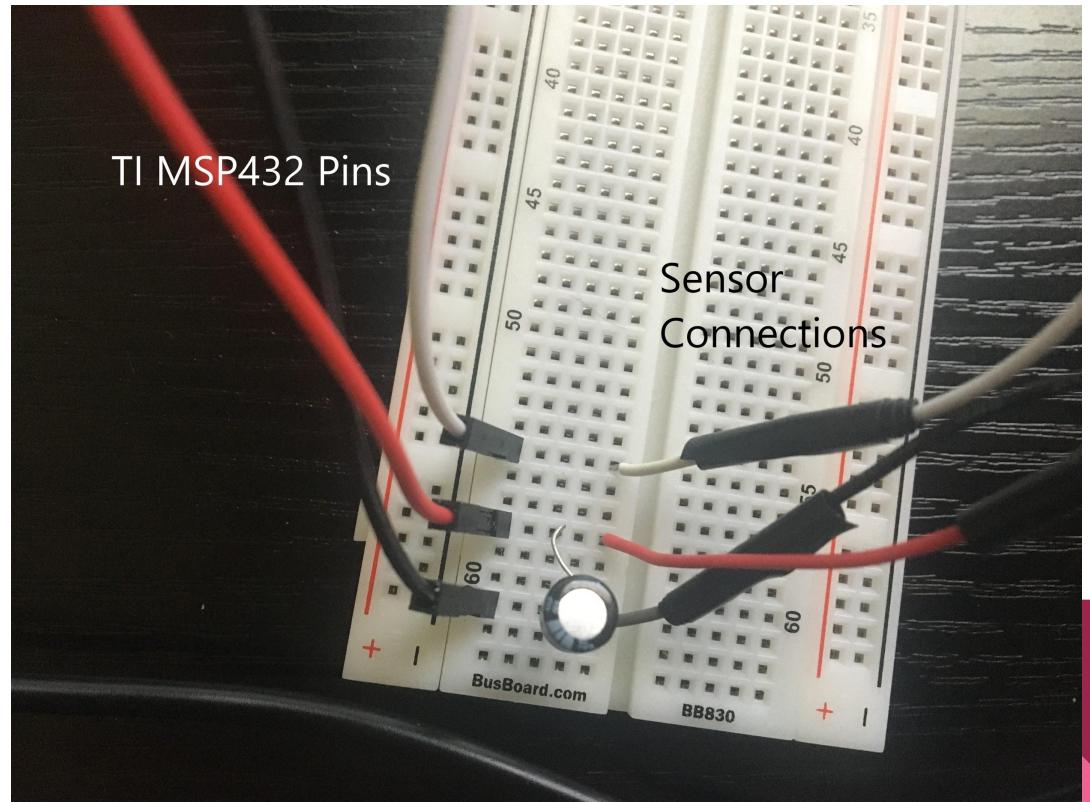


# TI MSP-EXP432 Pins 21-23



# Circuit

Circuit connections between sensor and MSP-EXP432 device, color coded appropriately. 10uF by-pass capacitor used to stabilize power supply line.



# C++ Code for MSP432

In programming this device, pin 23 is used as the analog input. An integer variable is used to capture the analog value from this pin. The Serial library prints this value every half second. The baud rate is set for 9600 bits/seconds.

EE437Team6ProjectUSBOut

```
// Code by Rishi Zaveri
// Subject: EE437
// 2 November 2020

int analogPin = 23; // Pin 23 Analog Input
int analogValue = 0;

void setup() {
    Serial.begin(9600);
}

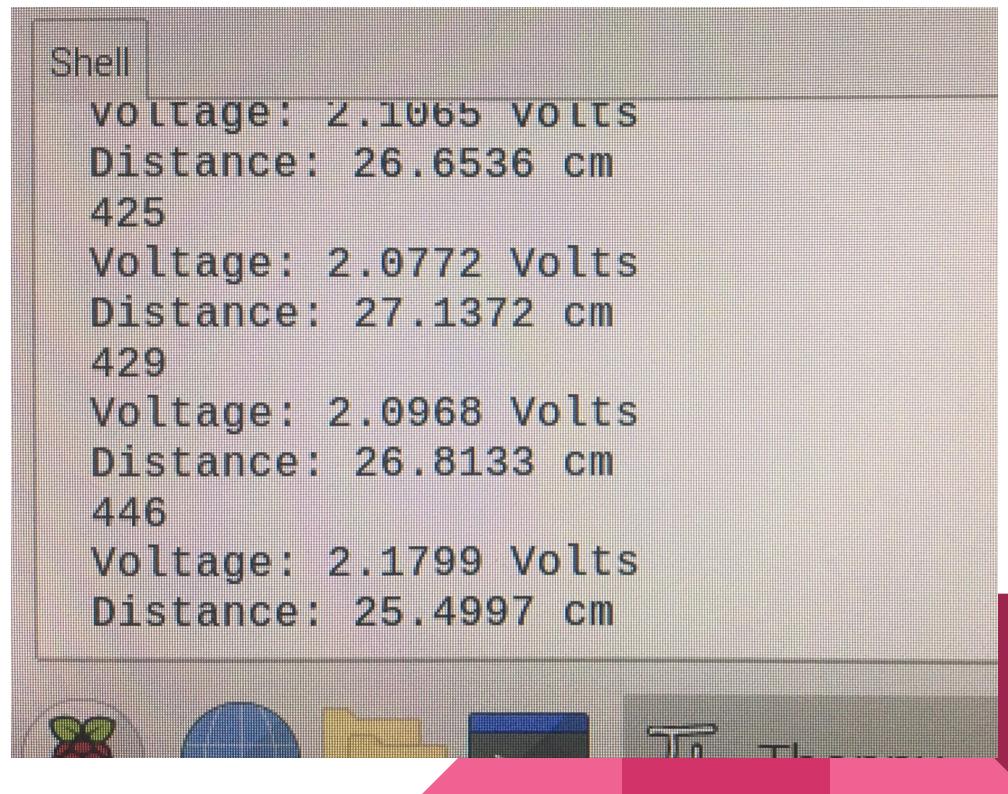
void loop() {
    analogValue = analogRead(analogPin);
    Serial.println(analogValue);
    delay(500);
}
```

```
2 # Produce sensor values
3 sensorValue = " "
4 voltageValue = 0.0
5 ser = serial.Serial('/dev/ttyACM0')
6
7 def sensing():
8     while True:
9         sensorValue = ser.readline()
10        sensorValue = sensorValue.decode('utf-8').strip()
11        print(sensorValue)
12        voltageValue = int(sensorValue) * (5.0/1023.0)
13        print("Voltage: " + str(round(voltageValue, 4)) + " Volts")
14
15        cmDistance = 10650.08 * pow(int(sensorValue), -0.935) - 10
16        print("Distance: " + str(round(cmDistance, 4)) + " cm")
17
18 t1 = threading.Thread(target=sensing)
19 t1.daemon = True # daemon thread terminates when program exits
20 t1.start()
```

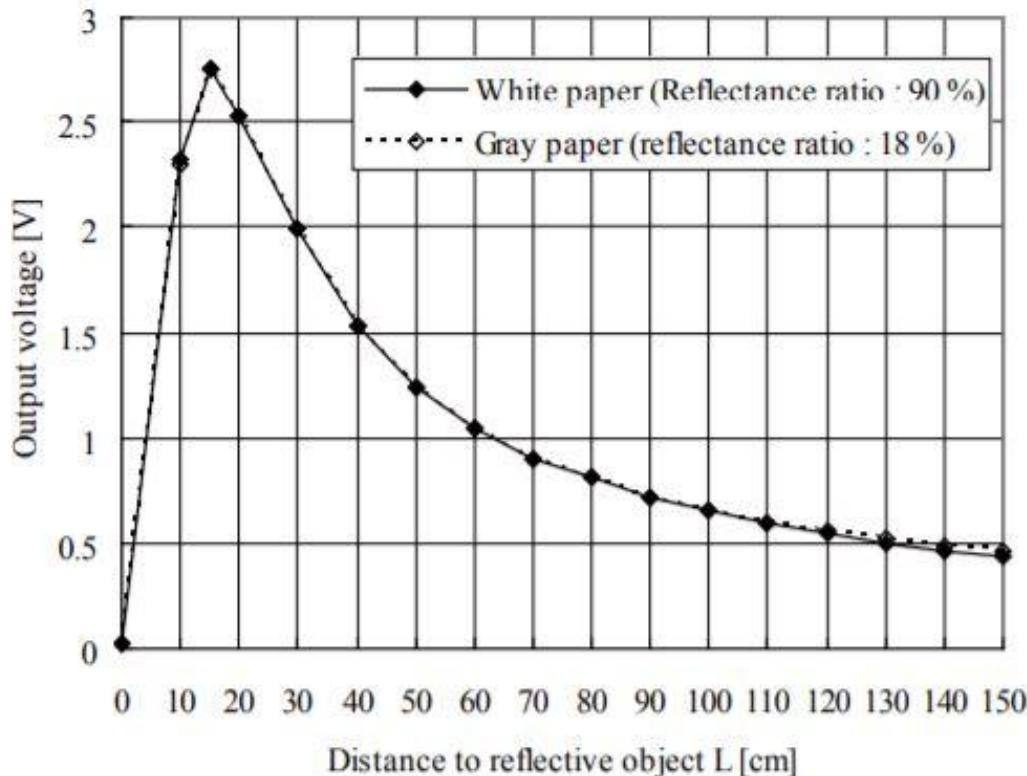
# Data Collection

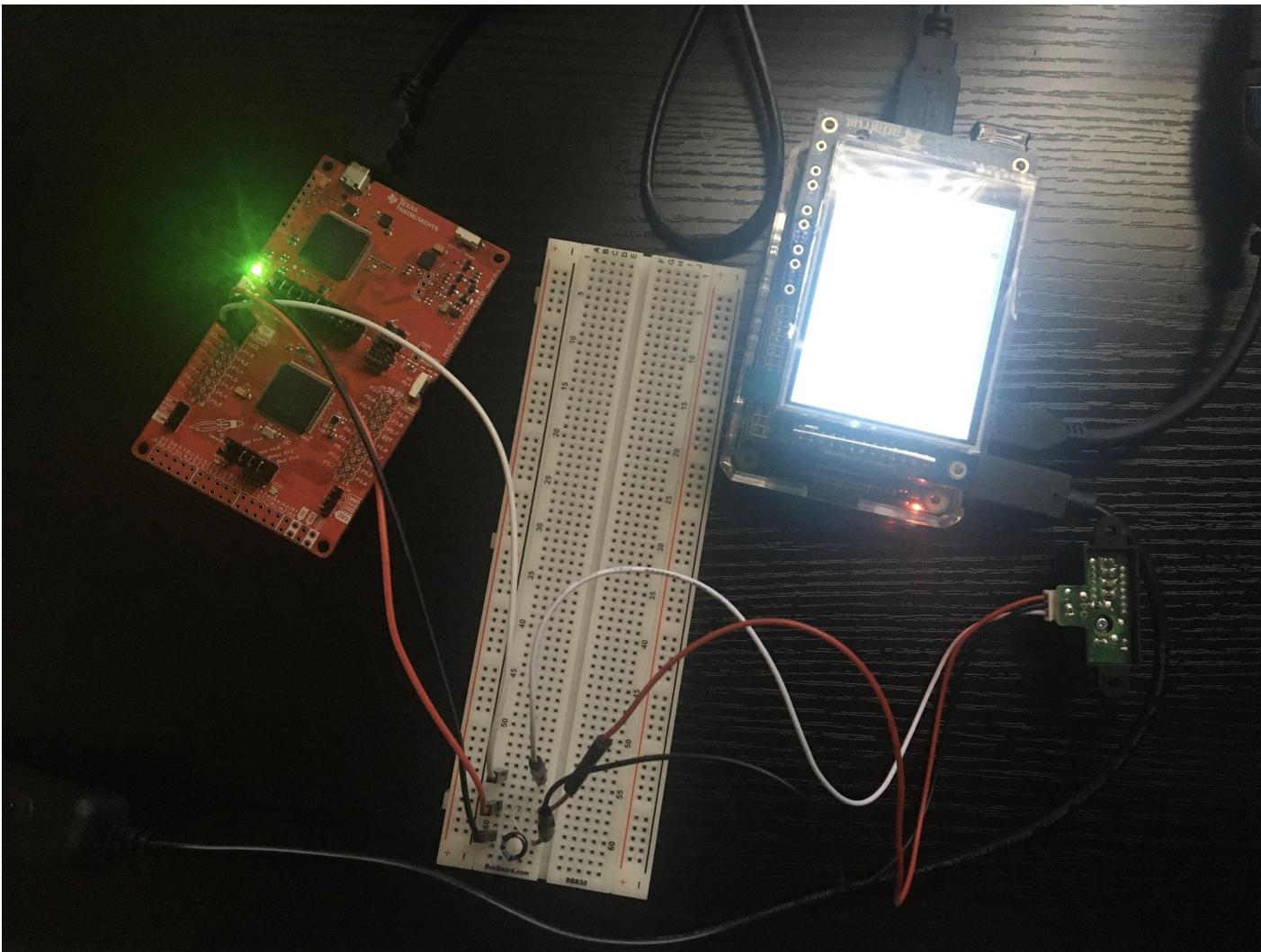
While the program is running, the Raspberry Pi collects information from the sensor in digital form. The data is then processed into the more useful voltage and distance values.

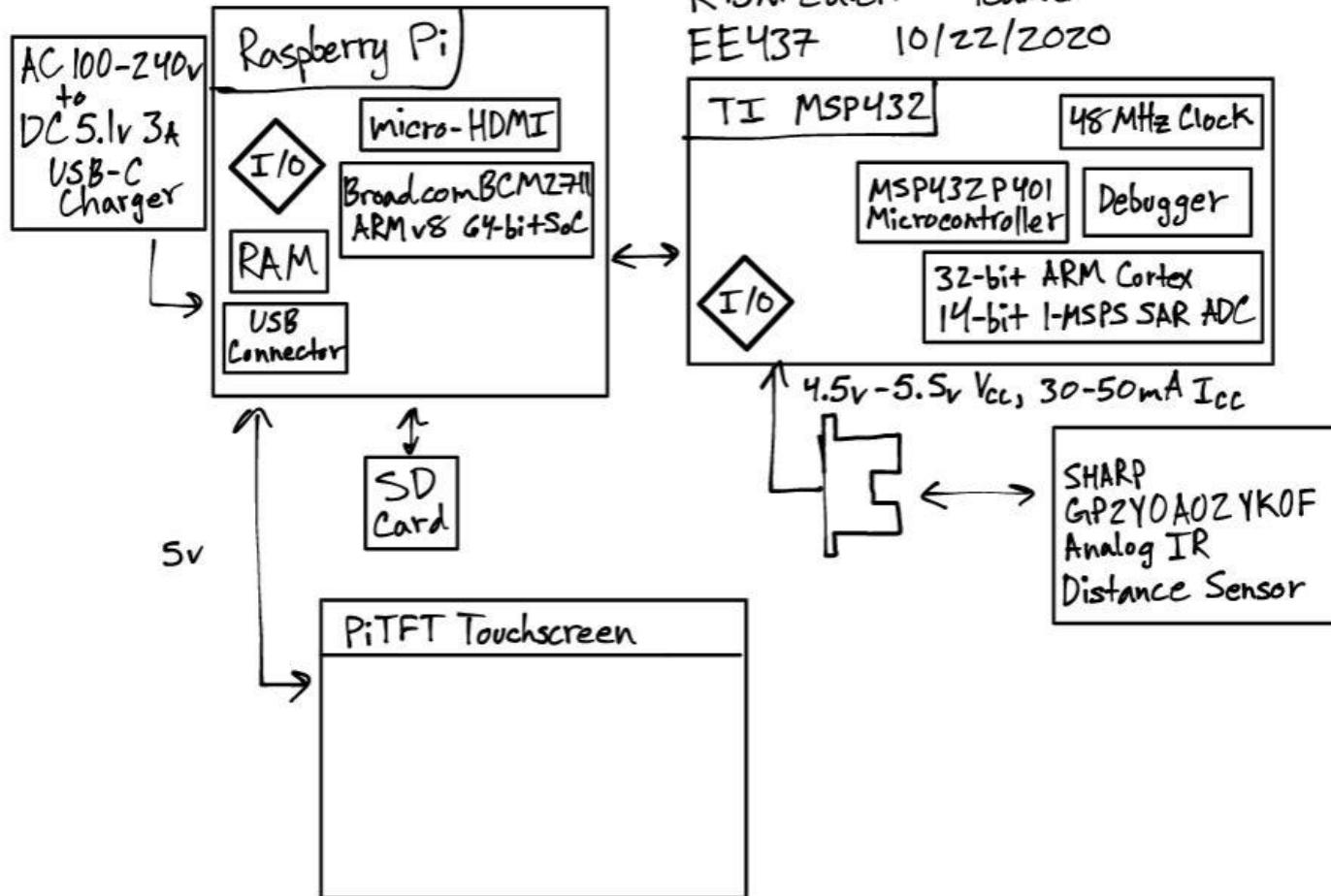
```
Shell
voltage: 2.1065 Volts
Distance: 26.6536 cm
425
Voltage: 2.0772 Volts
Distance: 27.1372 cm
429
Voltage: 2.0968 Volts
Distance: 26.8133 cm
446
Voltage: 2.1799 Volts
Distance: 25.4997 cm
```



# IR Sensor Distance-Voltage Relationship







Rishi Zaveri Team 6  
EE437 10/22/2020

| Rishi Zaveri, Team 6 |                              | Bill of Materials |
|----------------------|------------------------------|-------------------|
|                      | Part                         | Qty               |
| <b>1</b>             | TI MSP-EXP432P401R with      | 1                 |
| <b>2</b>             | Raspberry Pi 4 Model B       | 1                 |
| <b>3</b>             | SD Card with Raspbian OS     | 1                 |
| <b>4</b>             | SHARP GP2Y0A02YK0F IR Sensor | 1                 |
| <b>5</b>             | Breadboard                   | 1                 |
| <b>6</b>             | Raspberry Pi Power Supply 5V | 1                 |
| <b>7</b>             | Pi TFT 2.8" Touchscreen      | 1                 |
| <b>8</b>             | Jumper Wires Male/Female     | 3                 |
| <b>9</b>             | 10 microFarad Capacitor      | 1                 |

The background features a large, solid dark blue rectangle. In the upper right corner, there is an abstract geometric pattern composed of several triangles. These triangles are primarily in shades of blue, ranging from dark navy to light lavender. They are arranged in a way that creates a sense of depth and movement, resembling a stylized sunburst or a cluster of stars.

End