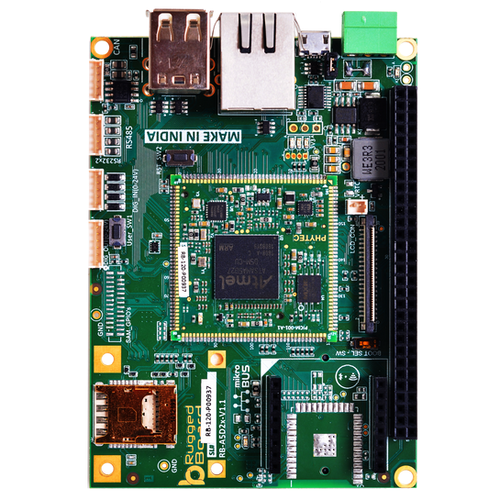
# M E T A L T O U C H S E N S O R

**RUGGED BOARD**

****

### METAL TOUCH SENSOR

### 

### Presented By, **SAISHITHA**

### Table of contents:

### 1.Abstract

1. **Hardware Components**

Metal Touch Sensor

Rugged Board-a5d2x

### Source code

1. **Pin Configurations 5.Connections**

**6.Conclusion**

1. **Abstract**

This research introduces a novel approach to human-machine interaction through the development of a Metal Touch Sensor with a rugged board, designed for applications in harsh environments. The proposed sensor system combines the sensitivity of touch-based interfaces with the durability of a rugged board, catering to industries such as automotive, industrial automation, and outdoor electronics.

The metal touch sensor employs capacitive touch technology, providing a reliable and responsive interface for users. The rugged board, constructed using advanced materials and engineering techniques, enhances the sensor's resilience to environmental factors such as moisture, temperature fluctuations, and mechanical stress. This ensures the sensor's reliable performance in challenging conditions, where conventional touch interfaces might fail.

### Hardware Components

1. Metal Touch Sensor.
2. Rugged Board-a5d2x.

### METAL TOUCH SENSOR:

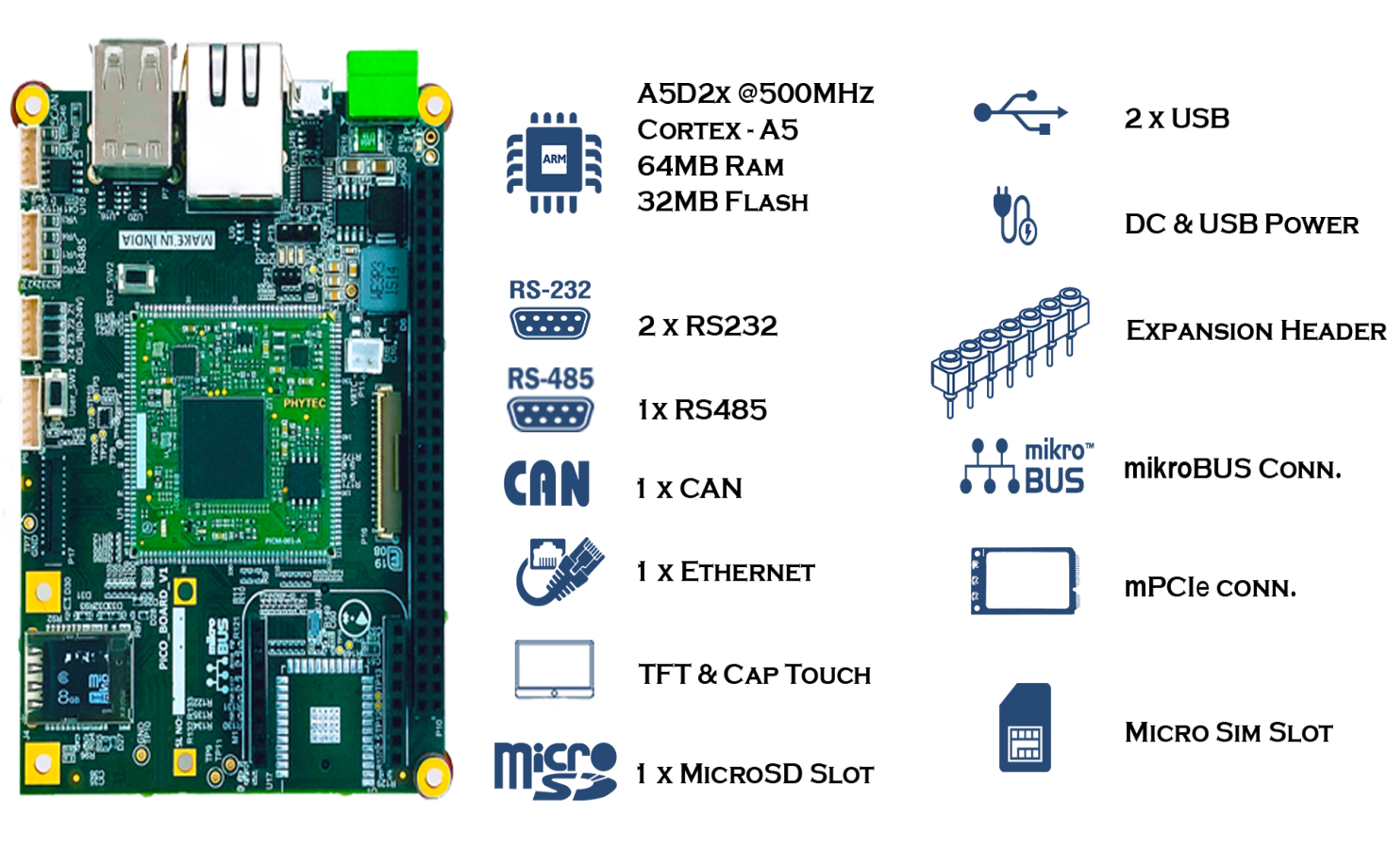
Principle of Operation:

Metal touch sensors operate based on capacitive touch technology. The principle of operation involves the use of capacitance changes to detect the presence or touch of a conductive object, such as a human finger. Here's a step-by-step explanation of the principle of operation for a metal touch sensor:

1. Capacitive Sensing Element:
   * The metal touch sensor consists of a capacitive sensing element, often in the form of a metal electrode or a grid of electrodes.
   * The sensing element is connected to a circuit that generates an electric field.
2. Electric Field Generation:
   * When the sensor is powered, it generates an electric field around the capacitive sensing element.
   * This electric field extends beyond the surface of the sensor.
3. Baseline Capacitance:
   * In the absence of any conductive object (touch), the sensor measures a baseline capacitance. This baseline capacitance is influenced by the characteristics of the sensing element and the surrounding environment.
4. Touch Detection:
   * When a conductive object, such as a human finger, approaches or touches the metal surface, it becomes part of the sensor's electric field.
   * The presence of the conductive object alters the capacitance of the sensing element. The capacitance increases due to the added capacitance of the conductive object.
5. Capacitance Change Detection:
   * The sensor's circuit continuously monitors the capacitance of the sensing element.
   * When a touch occurs, the change in capacitance is detected by the circuit.
6. Signal Processing:
   * The detected change in capacitance is then processed by the sensor's electronics. This processing involves analyzing the amplitude and frequency of the capacitance change.
7. Output Signal:
   * Based on the analysis, the sensor generates an output signal indicating the touch event. This output signal can be used to trigger specific actions, such as activating a button or initiating a command in a device.
8. Filtering and Noise Reduction:
   * To improve reliability, metal touch sensors often incorporate filtering and noise reduction techniques to distinguish genuine touch events from noise and interference.

### Rugged Board-a5d2x

Rugged Board is an Open Source Hardware & Software initiative to align with the fast growing Semiconductor technologies with a switch from classic to modern product development strategy & process. The usage of System on Module over a System on Chip is the rapid way to achieve time to market, curtail development risks for product quantities ranging from few to thousands.Rugged Board team targets to combine the Open source (Carrier Boards) community strength with industrial grade some and initiated the first Open Source Hardware "Industrial Pico Computer" which is powered by phyCORE-A5D2 SOM with Microchip A5D2x Cortex-A5 Core @500 Mhz.

 Fig :Rugged Board a5d2x

**3.Source code**

#include <stdio.h>

#include <stdlib.h>

#include <fcntl.h>

#include <unistd.h>

#include <string.h>

#define GPIO\_PIN\_PATH "/sys/class/gpio/PA31" // Replace with the actual GPIO path for PA31

#define LED\_PIN\_PATH "/sys/class/gpio/PC13" // Replace with the actual GPIO path for PC13

void error(const char \*msg) {

perror(msg);

exit(1);

}

void exportGPIO(const char \*gpioPath) {

int exportFile = open("/sys/class/gpio/export", O\_WRONLY);

if (exportFile == -1) {

error("Error exporting GPIO pin");

}

dprintf(exportFile, "%s", gpioPath + strlen("/sys/class/gpio/"));

close(exportFile);

}

void setDirection(const char \*gpioPath, const char \*direction) {

char directionPath[50];

snprintf(directionPath, sizeof(directionPath), "%s/direction", gpioPath);

int directionFile = open(directionPath, O\_WRONLY);

if (directionFile == -1) {

error("Error setting direction for GPIO pin");

}

dprintf(directionFile, "%s", direction);

close(directionFile);

}

void writeValue(const char \*gpioPath, const char \*value) {

char valuePath[50];

snprintf(valuePath, sizeof(valuePath), "%s/value", gpioPath);

int valueFile = open(valuePath, O\_WRONLY);

if (valueFile == -1) {

error("Error writing GPIO value");

}

dprintf(valueFile, "%s", value);

close(valueFile);

}

int readValue(const char \*gpioPath) {

char valuePath[50];

snprintf(valuePath, sizeof(valuePath), "%s/value", gpioPath);

int valueFile = open(valuePath, O\_RDONLY);

if (valueFile == -1) {

error("Error reading GPIO value");

}

char value;

read(valueFile, &value, 1);

close(valueFile);

return (value == '1') ? 1 : 0;

}

int main() {

// Export GPIO pins

exportGPIO(GPIO\_PIN\_PATH);

exportGPIO(LED\_PIN\_PATH);

// Set GPIO directions

setDirection(GPIO\_PIN\_PATH, "in");

setDirection(LED\_PIN\_PATH, "out");

while (1) {

// Read the GPIO value (0 or 1)

int touchValue = readValue(GPIO\_PIN\_PATH);

if (touchValue) {

printf("Touch detected\n");

writeValue(LED\_PIN\_PATH, "0"); // Turn on LED

} else {

printf("Not detected\n");

writeValue(LED\_PIN\_PATH, "1"); // Turn off LED

}

sleep(1); // Adjust the sleep duration as needed

}

// Unexport GPIO pins when done

writeValue(LED\_PIN\_PATH, "0"); // Turn off LED before unexporting

exportGPIO(GPIO\_PIN\_PATH);

exportGPIO(LED\_PIN\_PATH);

return 0;

}

## USING MRAA LIBRARY

## #include <stdio.h>

#include <unistd.h>

#include <mraa/gpio.h>

#define GPIO\_PIN\_NUMBER 41 // Replace with the actual GPIO pin number for PA31

#define LED\_PIN\_NUMBER 61 // Replace with the actual GPIO pin number for PC13

int main() {

mraa\_init();

// Create MRAA GPIO context

mraa\_gpio\_context gpioPin = mraa\_gpio\_init(GPIO\_PIN\_NUMBER);

mraa\_gpio\_context ledPin = mraa\_gpio\_init(LED\_PIN\_NUMBER);

if (gpioPin == NULL || ledPin == NULL) {

fprintf(stderr, "Failed to initialize GPIO pins\n");

return 1;

}

// Set GPIO directions

mraa\_gpio\_dir(gpioPin, MRAA\_GPIO\_IN);

mraa\_gpio\_dir(ledPin, MRAA\_GPIO\_OUT);

while (1) {

// Read the GPIO value (0 or 1)

int touchValue = mraa\_gpio\_read(gpioPin);

if (touchValue) {

printf("Touch detected\n");

mraa\_gpio\_write(ledPin, 0); // Turn on LED

} else {

printf("Not detected\n");

mraa\_gpio\_write(ledPin, 1); // Turn off LED

}

sleep(1); // Adjust the sleep duration as needed

}

// Cleanup and close GPIO pins when done

mraa\_gpio\_write(ledPin, 0); // Turn off LED before cleanup

mraa\_gpio\_close(gpioPin);

mraa\_gpio\_close(ledPin);

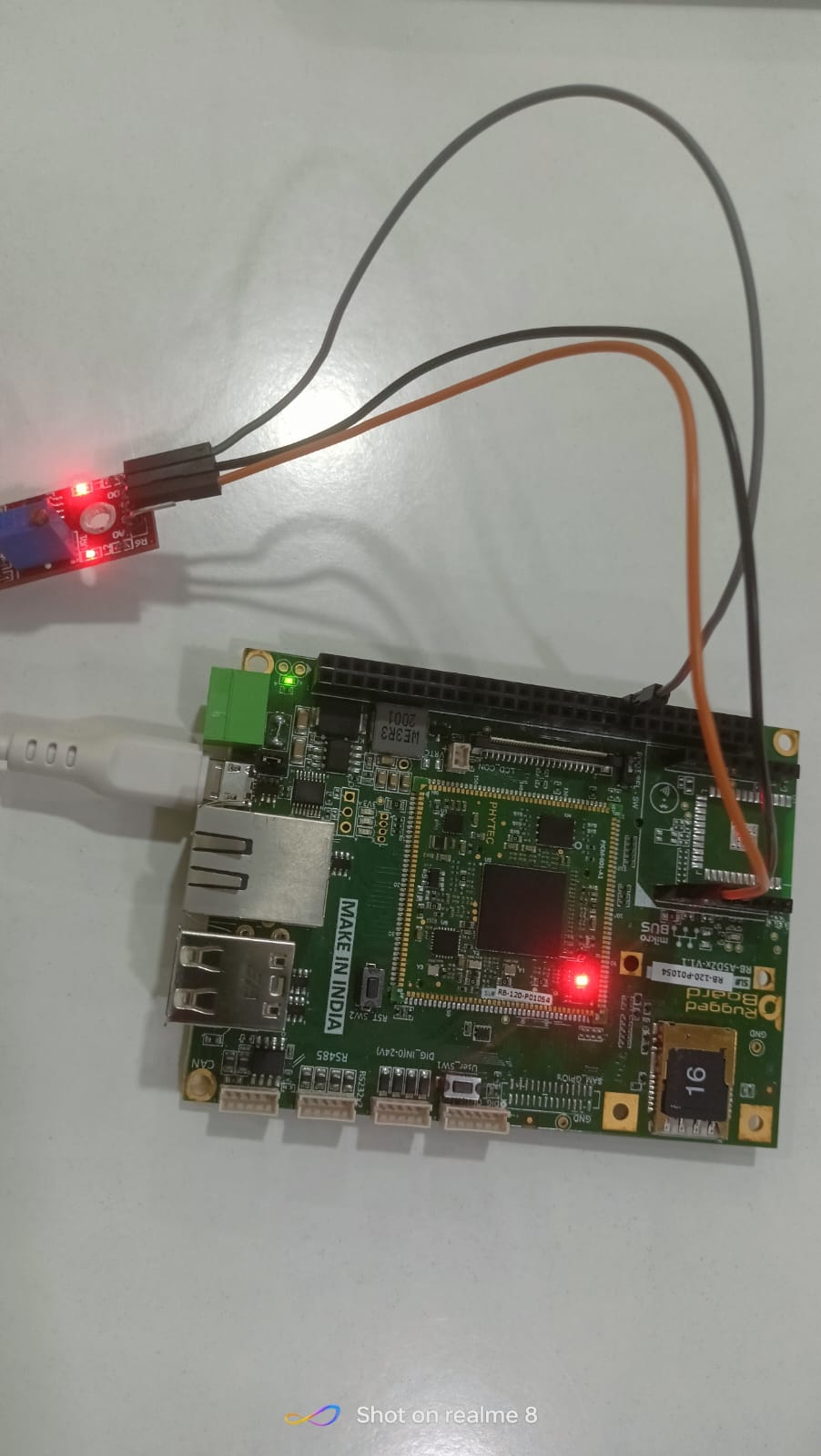
mraa\_deinit();

return 0;

}

## Pin Configurations

* Connect the VCC of sensor to the 5V of Rugged board in micro bus.
* Connect the GND of sensor to the GND of Rugged board.
* Connect digital pin of sensor to digital pin of Rugged board in micro bus.
* Set digital pin PA36 in a gpio extension and connect to the digital pin in the sensor.

1. **Connections**

**5.Conclusion**

The Metal Touch Sensor can be effectively interfaced with the Rugged Board A5D2x using the SYS library and MRAA library for GPIO access.

In conclusion, the metal touch sensor's capacitive sensing element, coupled with the rugged board's robust construction, ensures reliable operation in harsh conditions. The capacitive touch technology enables precise and responsive user interaction without the need for physical buttons, offering a modern and user-friendly interface. Meanwhile, the rugged board's resilience to environmental factors such as moisture, temperature variations, and mechanical stress enhances the sensor's longevity and suitability for demanding applications.