HACKATHON 1.0 - TEAM MAGIC SMOKERS

Hackathon Event https://events.pcbcupid.com/hackathon/1/instructions/

Abstract, Idea generation

What is this full of?

Absolutely nothing? No! This is a compact, modular desk gadget designed not just to boost productivity and discipline during study or work, but also to improve personal comfort and wellness. It combines time-management, environmental monitoring, user interaction, and smart notifications, all configurable via an app or web dashboard.

Hardware-Driven Features

- OLED Display (SH1106): High-contrast interface shows session times, progress, study stats, hydration alerts, and environmental readings.
- Tactile Controls: push buttons and potentiometers provide direct menu navigation, mode-switching, and real-time adjustment of parameters like timer durations.
- Customizable Alarms: Multiple alarms and reminders via buzzer (audio) and RGB/LEDs (visual), settable from the device or remotely.
- Temperature & Humidity Monitoring: DHT22 sensor tracks room conditions. Combined with hydration algorithms to send personalized water-reminder notifications.
- Session Feedback: LEDs (RGB, green, red) and a light bar visually indicate study/break phase, environmental status, or alerts at a glance.
- Multi-Mode Operation: Study Mode, Relax Mode, Hydration Mode, and Custom routines—all switchable from device buttons or over Bluetooth/Wi-Fi via the app.
- Productivity Screen: Tracks Pomodoro cycles, completed tasks, time spent, and session analytics on the web dashboard.

Connectivity & Configuration

- App/Web Dashboard: Set timers, manage alarms, view stats and sensor data, control recording, and update firmware from your phone or PC.
- Wireless Sync to web dashboard: WiFi local server (via Glyph C3) and remote configuration.

Extra Perks

- Offline Operation: Core features work standalone—no internet needed, ensuring distraction-free and privacy.
- Hands-On Customization: Designed for circuit enthusiasts—expose test points, debugging headers, and I/O for easier tinkering, firmware updates, and upgrades.
- Motivational Focus: Wellness nudges for hydration and posture, extra-motivational messaging for productivity, and a visual “focus progress” LED bar.

List of hardware components used

Sl. No	Component	Quantity	Price Per Unit (INR)	Sumup Price (INR)
1	Glyph C3(headers not soldered)	1	434	434
2	FR4 – High Quality Zero board PCB (Perfboard)	3	35	105
3	OLED Display (SH1106)	1	220	220

4	Buzzer	1	15	15
5	Tactile Push Button Switch Self Lock(DPdT Switch)	1	10	10
6	Right angle tactile push button	6	5	30
7	RGB LED	7	10	70
8	5mm Green LED	2	3	6
9	5mm Red LED	2	3	6
10	Resistor Box	1	70	70
11	Potentiometer	2	25	50
12	DHT 22 - Humidity and Temperature Sensor	1	199	199
13	Ultrasonic Sensor	1	112	112
14	LDR(5mm)	1	20	20
15	Female header pins	5	10	50
16	Male header pins	4	10	40
17	Wires 1Meter	4	15	60
			Total (INR)	1497
			Saved (INR)	503

Pin configuration and circuit connections

Circuit design/schematic diagrams

Controller Pinout

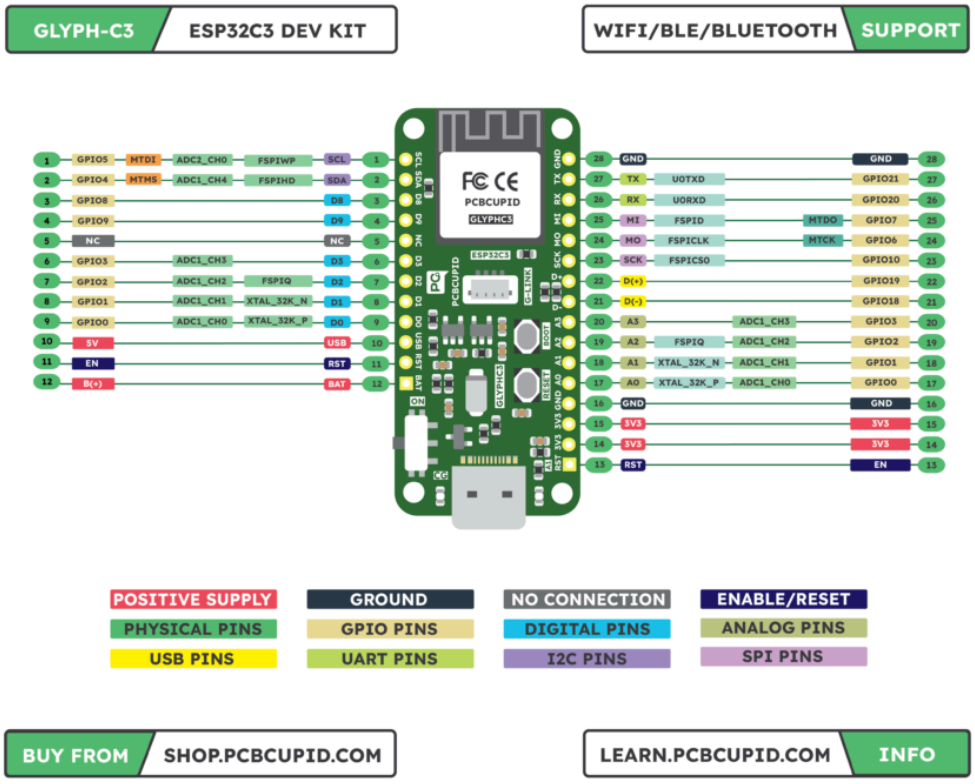


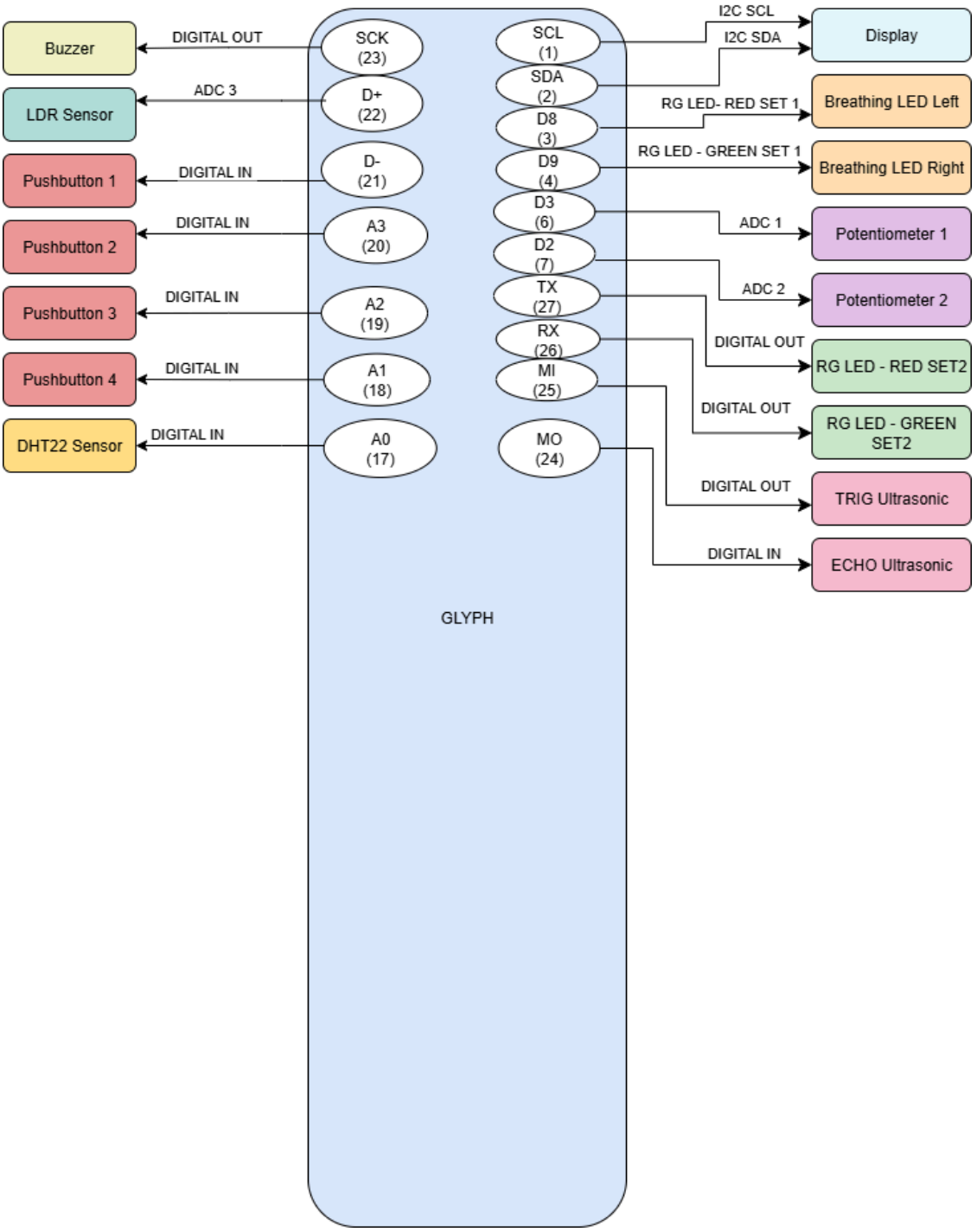
Figure: Glyph C3 Pinout Diagram

Pins and Peripherals Used

GLYPH Pin	Pin Number	Peripheral	Application
SCL	1	I2C SCL	Display
SDA	2	I2C SDA	
D8	3	RG LED - RED SET1	Breathing LED Left
D9	4	RG LED - GREEN SET1	Breathing LED Right
D3	6	ADC 1	Potentiometer 1
D2	7	ADC 2	Potentiometer 2
TX	27	Digital Out	Breathing LED - RED SET2

RX	26	Digital Out	Breathing LED - GREEN SET2
MI	25	Digital Out	TRIG Ultrasonic (5V VCC)
MO	24	Digital IN	ECHO Ultrasonic (5V VCC)
SCK	23	Digital Out	Buzzer
D+	22	ADC 3	LDR Sensor
D-	21	Digital IN	Pushbutton 4 (Spare)
A3	20	Digital IN	Pushbutton 3
A2	19	Digital IN	Pushbutton 2
A1	18	Digital IN	Pushbutton 1
A0	17	Digital IN	DHT22 Sensor

PIN MAPPING



Code (with comments, if applicable)

Development Flow

1. Show DHT22 values on the OLED display and keep showing it on the top side of the display. It is connected on pin A0.
2. Import current time and display live clock as well through NTP via WiFi connection.

3. On clicking Button 1, change the clock to a 20 minute countdown timer, then Button 4 to reset and go back to the live clock. Beep buzzer for 1 second at the start of timer, stop of timer and reset of timer.
4. Turn on RG LEDs in breathing mode (PWM) RG SET1 and RG SET2 on clicking Button 2, and Button 3 respectively. If pressed again, turn the breathing LED off.
5. All the functions work when a person is present and active as detected by ultrasound sensor TRIG and ECHO pins. Take the device to sleep mode and display Sleeping… zzz when the ultrasonic sensor does not detect anyone in front and keep the RG LEDs in breathing mode forever.

```
#include <Arduino.h>
#include <WiFi.h>
#include <NTPClient.h>
#include <WiFiUdp.h>
#include <U8g2lib.h>
#include <Wire.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

// WiFi credentials
const char* ssid = "Magicsmokers";
const char* password = "12348765";

// DHT22 configuration
#define DHTPIN A0          // Change to your DHT22 data pin
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

// Initialize SH1106 OLED display
U8G2_SH1106_128X64_NONAME_F_HW_I2C u8g2(U8G2_R0, U8X8_PIN_NONE);

// NTP Client Setup
WiFiUDP ntpUDP;
NTPClient timeClient(ntpUDP, "pool.ntp.org", 19800, 60000); // GMT+5:30 (IST)
offset in seconds, update interval 60s

// LED pins (active low)
const uint8_t LED_RED_RIGHT = 3;   // D8 Pin 3
const uint8_t LED_GREEN_RIGHT = 4; // D9 Pin 4
const uint8_t LED_RED_LEFT = 27;   // TX Pin 27
const uint8_t LED_GREEN_LEFT = 26; // RX Pin 26

// Breathing parameters
const uint8_t NUM_LEDS = 4;
```

```
const uint8_t ledPins[NUM_LEDS] = {LED_RED_RIGHT, LED_GREEN_RIGHT, LED_RED_LEFT,
LED_GREEN_LEFT};

// PWM brightness range for active low LEDs (0 = fully on, 255 = off)
// We'll map a sine wave or linear fade to pwm values in reverse
// Use brightness from 0..255 for ease, invert in analogWrite
const int breatheSteps = 100; // number of steps to fade in/out
const unsigned long breatheInterval = 10;  // ms duration between steps

uint8_t currentLedIndex = 0;
int breatheStep = 0;
bool breatheIncreasing = true;
unsigned long lastBreatheTime = 0;

void setup() {
 Serial.begin(115200);

 // Initialize DHT sensor
 dht.begin();

 // Initialize OLED
 u8g2.begin();

 // Initialize LED pins as output and turn them off (HIGH since active low)
 for (uint8_t i = 0; i < NUM_LEDS; i++) {
   pinMode(ledPins[i], OUTPUT);
   digitalWrite(ledPins[i], HIGH);
 }

 // Connect to WiFi
 WiFi.begin(ssid, password);
 Serial.print("Connecting to WiFi");
 while (WiFi.status() != WL_CONNECTED) {
   delay(500);
   Serial.print(".");
 }
 Serial.println("\nWiFi connected");

 // Initialize NTP
 timeClient.begin();
 timeClient.update();
}

void breatheLED() {
 unsigned long currentMillis = millis();
```

```
 if (currentMillis - lastBreatheTime < breatheInterval) {
   return; // not time to update yet
 }
 lastBreatheTime = currentMillis;

 // Calculate brightness (0-255) peak to off, active low means invert PWM value
 int brightness;
  // Use a smoother brightness curve based on sine for nicer breathing
 float pi = 3.14159265;
 float brightnessFactor = (1 - cos(pi * (float)breatheStep / breatheSteps)) / 2.0;
 // brightnessFactor goes 0->1->0 in gradual breathing

 brightness = (int)(brightnessFactor * 255);

 // Because LEDs are active low, invert brightness for analogWrite
 uint8_t pwmValue = 255 - brightness;

 // Write PWM to current LED
 // Turn off all other LEDs
 for (uint8_t i = 0; i < NUM_LEDS; i++) {
   if (i == currentLedIndex) {
     analogWrite(ledPins[i], pwmValue);
   } else {
     digitalWrite(ledPins[i], HIGH); // off
   }
 }

 // Update breathe step
 breatheStep++;
 if (breatheStep > breatheSteps) {
   breatheStep = 0;
   // Move to next LED
   currentLedIndex++;
   if (currentLedIndex >= NUM_LEDS) {
     currentLedIndex = 0;
   }
 }
}

void loop() {
 // Update time from NTP periodically
 timeClient.update();

 // Read temperature and humidity
 float temperature = dht.readTemperature();
```

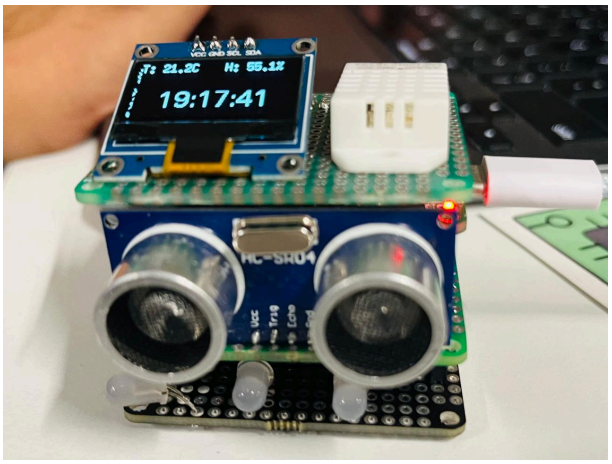
```
 float humidity = dht.readHumidity();

 // Clear display buffer
 u8g2.clearBuffer();

 // Set font for sensor data
 u8g2.setFont(u8g2_font_6x13_tr);

 // Display temperature and humidity at the top
 char tempStr[16];
 char humStr[16];
 if (isnan(temperature) || isnan(humidity)) {
   u8g2.drawStr(0, 12, "Sensor Error!");
 } else {
   snprintf(tempStr, sizeof(tempStr), "T: %.1fC", temperature);
   snprintf(humStr, sizeof(humStr), "H: %.1f%%", humidity);
   u8g2.drawStr(0, 12, tempStr);
   u8g2.drawStr(70, 12, humStr);
 }

 // Prepare time string (HH:MM:SS)
 unsigned long epochTime = timeClient.getEpochTime();
 int currentHour = (epochTime % 86400L) / 3600;  // hour in 24h format
 int currentMinute = (epochTime % 3600) / 60;
 int currentSecond = epochTime % 60;

 char timeStr[16];
  snprintf(timeStr,  sizeof(timeStr),  "%02d:%02d:%02d",  currentHour,  currentMinute,
currentSecond);

 // Set larger font for center time display
 u8g2.setFont(u8g2_font_fur17_tr);

 // Calculate x position to center the time string (128 width display)
 int16_t textWidth = u8g2.getStrWidth(timeStr);
 int16_t xPos = (128 - textWidth) / 2;

 // Draw time centered vertically and horizontally (place vertically near center)
 u8g2.drawStr(xPos, 54, timeStr);

 // Send buffer to display
 u8g2.sendBuffer();

 // Update breathing LEDs
 breatheLED();
```

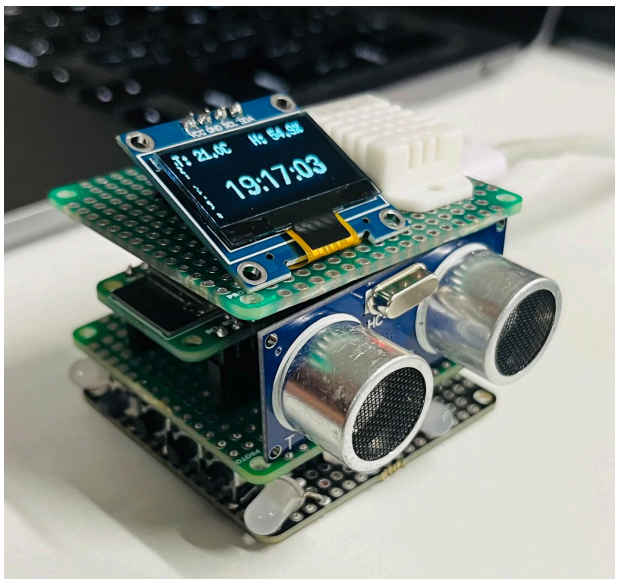
```
 // Delay removed or reduced to allow smooth breathing effect
 // Smaller delay to allow for breathing effect updates smoothly
 delay(10);
}
```

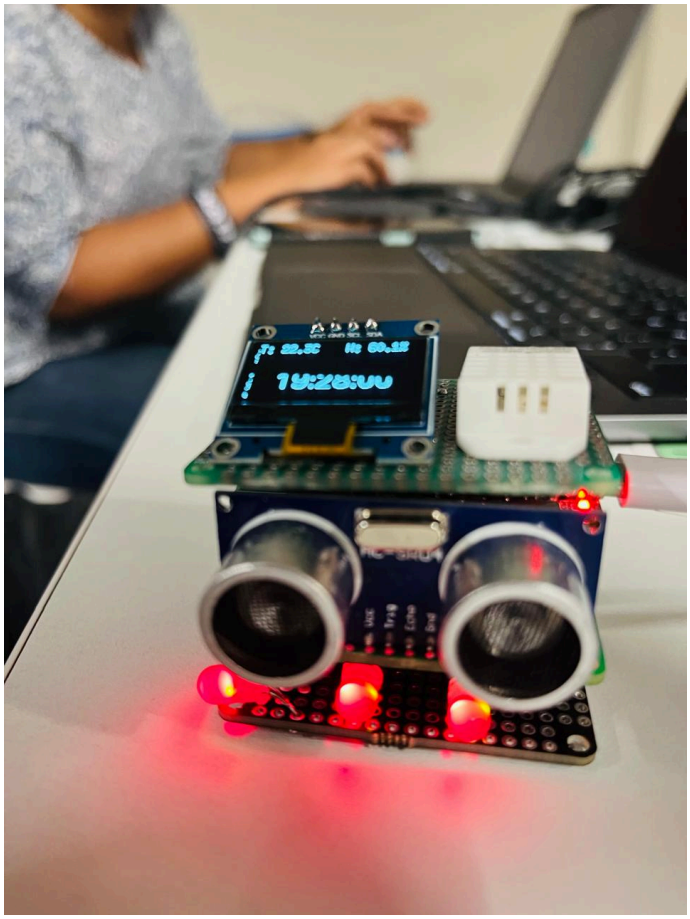
Software/tools used

Arduino IDE

Clear pictures of the setup/prototype







PROTOTYPE SHOWING THE TEMPERATURE,HUMIDITY AND TIMER

Working explanation and output

Applications and other relevant details

Applications

- **Student Study Aid**: Organizes focused sessions with timers and offers audio recording/playback for notes and lectures.
- **Professional Productivity**: Manages tasks, breaks, and multi-mode workflows via alarms and analytics.
- **Health Monitoring**: Tracks temperature and humidity to provide personalized hydration reminders.
- **Remote Connectivity**: Bluetooth/Wi-Fi enabled for app/web-based control, monitoring, and customization.
- **Time & Task Tracking**: Displays session progress, Pomodoro counts, and productivity stats on device and dashboard.
- **Customizable & Expandable**: Designed for makers with support for firmware updates and additional sensors/features.
- **Stress Relief Features**: Relax Mode with ambient lighting, soothing sounds, and motivational prompts to boost focus.
- **Ergonomic Integration**: Controls external devices (e.g., lamps) and supports potential ergonomic health reminders.

Target Customers of the Study Buddy Device

- **Students**:Seeking tools to improve focus, structure study time, and record notes effectively.
- **Working Professionals**:Needing to organize tasks, schedule breaks, and maintain productivity in office or remote work environments.
- **Remote Workers & Home Office Users:**Interested in smart, connected devices for customization and monitoring of their workspace comfort and efficiency.
- **Health-Conscious Individuals:**Prioritizing hydration, posture awareness, and balanced work-rest cycles to enhance wellness.
- **Productivity Enthusiasts:**Users of Pomodoro and similar techniques who value detailed tracking and motivation tools.
- **DIY Makers & Electronics Hobbyists:**Looking for devices with customization options, firmware update capability, and sensor expansion.
- **Educational Institutions:**Aiming to provide students with smart study aids to foster better learning habits and wellness monitoring.
- **Tech-Savvy Consumers:**Interested in IoT-enabled, app-configurable devices that integrate into existing smart home or work systems.

Ambient Lighting and Mood Enhancement – Future Scope

Abstract:

The lighting technology used for residential and commercial establishments has improved tremendously in recent years. From the era of incandescent lamps to the modern LED lighting systems, the transition is remarkable with power saving, better illumination, mood lighting thereby improving the user satisfaction. In this paper a localized strategy for LED lighting control is proposed, whereby available illumination is utilized judiciously. Additional integration of the control with the atmosphere in the room, no of persons occupying the room is also considered.

REFERENCE: 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1)