SESSION NAME	PROJECT: SAFE GUARD MONITOR (SGM)
Session Description	Students will learn to make safe guard monitor using LED, Buzzer & Ultrasonic sensor using text based coding.
Category	ARDUINO
Class Duration	60 mins
Objective	Learn to make SAFE GUARD MONITOR
Resources	Trainer Resources • Use an earphone and mic • Keep a notepad and pen • Use Crinnolabs app to mark attendance • Arduino IDE Software • DIY Kit Student Resources • Earphones with Mic • Notepad and Pen • Arduino IDE Software • DIY kit
Check of List	**Before The Session** • Edit batch timing • Give announcements (Including Time and Link) • Give message through app chat group (Including Link) *Due to any reason unable to conduct the session, then the trainer needs to update reason as a task status summary with rescheduling request / Inform technical head through call/WhatsApp immediately Technical head will review the request, then reschedule the session After rescheduling by technical head, trainer need to inform the students…..* *Important Note for Trainer:(If students complete the works before the session time ends,Trainer should make them engaged by giving the task which is on the PPT mentioned for next session. Engage them and complete the session on time (1Hr)* *Important Note for Trainer:* **Ask Students to share their feedback regarding the session. It should be in Crinnolabs application. They can share feedback once the attendance is marked.)* **Upload the study materials to the batch and inform them through app.* **On Session** • Student need to mute mic and turn on video • Trainer and student turn on screensharing and chat. • Take Attendance • Explain Task for student • Sum up session *Student and trainer need to share their screen and turn on camera entire the session*

PROJECT: SAFE GUARD MONITOR (SGM)

SGM Introduction	**0.1**
SGM Circuit Making	**0.2**
SGM Coding	**0.3**
SGM Summary of the Session	**0.4**

SGM 0.1 - Introduction

Hello, everyone! I'm **Name** from STEM-Xpert, and I'm here with you today to make this session an exciting learning experience!

Let's get started with our session.
Are you ready?
Great! Let's get started!"

(show output with background music)

Today, we'll be working on an exciting project called the *Safe Guard Monitor*!

(show ppt image2)

This project will help us dive into some important coding concepts, including:

- Variables – these are like containers where you can store information.
- Data Types – which help define the kind of information stored in those variables.
- The `#define` function – a handy way to assign names to values, making your code easier to read and maintain.
- The `delayMicroseconds()` function – which allows you to pause operations for tiny durations.
- And control structures like `if...else` – which let your program make decisions based on conditions.

Now, let's talk about the project itself. What is the *Safe Guard Monitor*?

(show video-https://youtu.be/vf2lW4LkmMQ?si=bRuGBde9I38b_Xo6)

It's an ultrasonic-based monitoring system. Imagine a system that can sense when something is close or far away. Here's how it works:

- The ultrasonic sensor detects the distance of an object.
- If an object comes close, the system activates—the buzzer beeps, and the LED lights up.
- As the object moves away, both the buzzer and the LED turn off automatically.

This system has practical applications. For example, you could use it to monitor security in a room by detecting intrusions or set it up for obstacle detection in robotics projects.

(Here, show the project working with visuals of the sensor, buzzer, and LED in action.)

Pretty interesting, right?

Next, let's go over the components you'll need to build this project. Make sure to take notes or gather these components if you haven't already. Here's the list:

1. One Arduino Uno board – this is the "brain" of your project.
2. One USB cable – to connect the Arduino to your computer.
3. One HC-SR04 Ultrasonic Sensor Module – this sensor measures distances by sending out ultrasonic waves and detecting their echoes.
4. One buzzer – it will beep when the sensor detects something nearby.
5. One LED – to visually indicate the detection.
6. Two resistors – a 330-ohm resistor and a 100-ohm resistor, which protect the LED and other components from too much current.
7. One breadboard – to assemble the circuit without soldering.
8. Eight male-to-male jumper cables – these help you connect the components easily.

Take a moment now to gather all the components we just discussed. Ensure you have everything you need before we move forward

I hope you've all gathered the components by now. If you have everything ready, that's fantastic—great job!

Let's recap what we've covered today:

We introduced the *Safe Guard Monitor* project—a simple yet powerful distance-monitoring system that uses an ultrasonic sensor, buzzer, and LED. We also discussed the coding concepts you'll learn and reviewed the components required to bring this project to life.

In our next session, we'll begin assembling the circuit and writing the code step by step. Get ready to explore how each component fits together and how to program the Arduino to make it all work!

Thank you for your attention, and I'll see you in the next session. Take care!

STEM-Xpert
NURTURING INNOVATION

SGM 0.2-Hardware/Circuit Making video

Welcome back, everyone! In our first session, we introduced the Safe Guard Monitor project and explored the components required for building it, including the Arduino Uno, ultrasonic sensor, buzzer, and LED. Now that we're familiar with the components, it's time to bring the project to life!

Let's get started with the hardware setup for the Safe Guard Monitor.

Follow along step by step as we bring this project to life.
Let's begin with the ultrasonic sensor, which has four pins: VCC, Ground, Trigger, and Echo.

- Connect the VCC pin to the 5 Volt pin on the Arduino to supply power.
- Connect the Ground pin to the ground pin of the Arduino.
- Next, connect the Trigger pin to pin 13 on the Arduino.
- Finally, connect the Echo pin to pin 12.
  Now, let's move on to the buzzer.
- Connect the positive terminal of the buzzer to pin 7 of the Arduino using a 100 ohm resistor.
- The negative terminal should be connected to the ground pin on the Arduino.

(make gap 1 sec each steps for making the connection)

The resistor here is essential to protect the buzzer from high current.

Next, we'll connect the LED, which serves as a visual indicator.

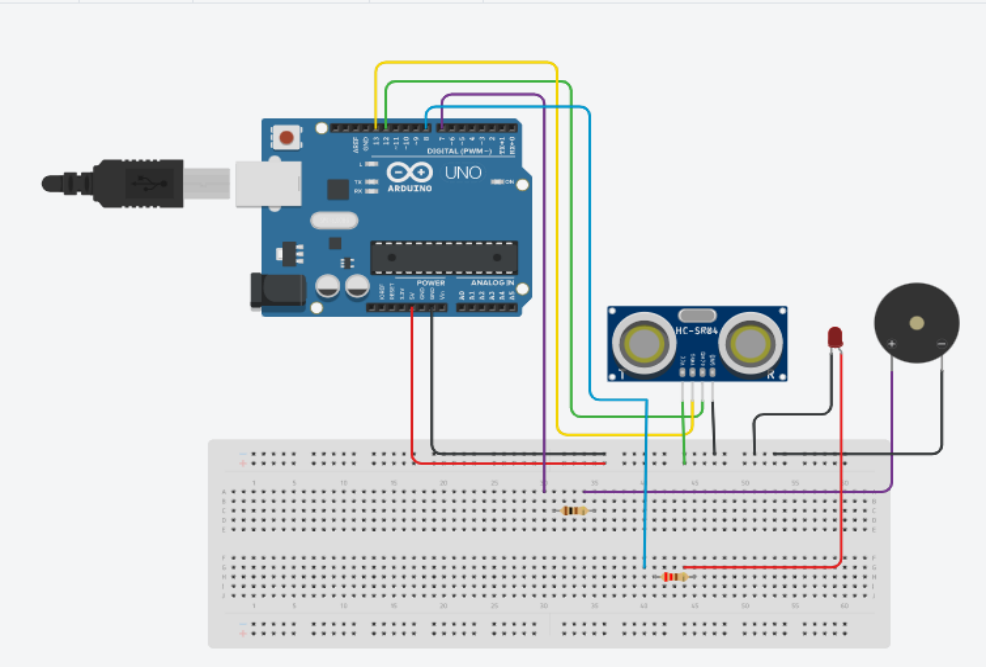
- Connect the longer leg of the LED, also known as the positive leg, to pin 8 of the Arduino using a 330 resistor.
- The shorter leg, or negative leg, should be connected to the ground pin.

The resistor limits current flow to protect the LED from damage.

Now that all the components are connected, take a moment to double-check everything against the circuit diagram. Accurate connections are crucial for the system to function correctly.

(Show circuit diagram Image 3)

Circuit Diagram



Great progress, everyone! We completed the hardware setup for our Safe Guard Monitor. We connected the ultrasonic sensor, buzzer, and LED to the Arduino, ensuring proper connections with resistors for safety. Double-checking the connections against the circuit diagram was a crucial step to ensure everything is ready. In the next session, we'll move on to coding to bring our project to life.

SGM 0.3- Coding

Welcome back, everyone! In our second session, we successfully completed the hardware setup for the Safe Guard Monitor. We connected the ultrasonic sensor, buzzer, and LED to the Arduino, ensuring proper connections and safety measures. Now, it's time to move on to the next exciting step, coding and testing our project!

Code

```
/* This simple project describes how to make an ultrasonic alarm system using
LED, Ultasonic Sensor(HC-SR04) and a buzzer.*/

//Firstly the connections of ultrasonic Sensor.Connect +5v and GND normally &trigger pin to 12 &
echo pin to 13.

#define trigPin 13
#define echoPin 12
int Buzzer = 7; // Connect buzzer pin to 7
int ledPin= 8;  //Connect LED pin to 8
int duration, distance; //to measure the distance and time taken
void setup() {

      //Define the output and input objects(devices)
      pinMode(trigPin, OUTPUT);
      pinMode(echoPin, INPUT);
      pinMode(Buzzer, OUTPUT);
      pinMode(ledPin, OUTPUT);
}
```

```
void loop() {
   digitalWrite(trigPin, HIGH);
   delayMicroseconds(10);
   digitalWrite(trigPin, LOW);
   duration = pulseIn(echoPin, HIGH);
   distance = (duration/2) / 29.1;
   //when distance is greater than or equal to 10 OR less than or equal to 0,the buzzer and LED are
off
  if (distance >= 10 || distance <= 0)
       {
       digitalWrite(Buzzer,LOW);
       digitalWrite(ledPin,LOW);
       }
  else {
       digitalWrite(ledPin,HIGH);
       digitalWrite(Buzzer,HIGH);

  }
}
```

Here is the simple code for reading the distance value from HC SR04 Ultrasonic Sensor using the Arduino Code.

"We start by defining the pins for the ultrasonic sensor, buzzer, and LED. The trigPin and echoPin are assigned to pins 13 and 12 respectively. The buzzer is connected to pin 7, and the LED is connected to pin 8. Additionally, we declare two variables, duration and distance, to calculate the time and distance measurements."

"In the setup() function, we configure the pins as input or output. The trigger pin and the pins for the LED and buzzer are set as outputs, while the echo pin is set as an input. This ensures proper communication with the connected components."

"In the loop() function, we send a high signal to the trigger pin for 10 microseconds. This generates an ultrasonic pulse from the sensor. After the pulse, we set the trigger pin to low to wait for the reflected signal."

"The pulseIn function listens for the echo signal, which returns the time it took for the ultrasonic wave to travel to the object and back. This time is stored in the duration variable."

"The distance is calculated using the formula distance = (duration divided by 2) divided by 29.1”

This converts the time into a distance in centimeters by considering the speed of sound and dividing the time by two for the round trip."

"Finally, the distance value determines the behavior of the system. If the distance is 10 centimeters or more, or if it’s zero, the LED and buzzer remain off. Otherwise, both the LED and buzzer are activated, signaling the detection of a nearby object."

To upload the code to the Arduino Board, Select the Arduino UNO Board from Tools Menu and also select the COM port. Then upload the code by clicking on upload button.

Once uploaded, the Arduino will process the code and control the components accordingly.
Now comes the exciting part testing your Safe Guard Monitor!

("Show the project's working/output.")

Observe the following:

- The buzzer should start beeping as the object approaches.
- Simultaneously, the LED should light up, providing a visual alert.
- When the object moves out of range, both the buzzer and LED should turn off.

In this session, we coded the Safe Guard Monitor to bring it to life. We defined pin assignments for the ultrasonic sensor, buzzer, and LED, and calculated the distance using the sensor's input. The code activates the LED and buzzer when an object is detected within 10 cm and turns them off otherwise.

Excellent progress, your Safe Guard Monitor is now complete!

SGM 0.4- Project Review

"Congratulations on completing the Safe Guard Monitor project! Let's take a moment to reflect on what we've learned.

In this project, we combined an ultrasonic sensor, LED, and buzzer to create a simple and effective security system. By connecting these components to the Arduino using a breadboard and wires, we built an organized setup.

The ultrasonic sensor acted as the system's eyes, sending sound waves, detecting reflections, and measuring distance. With the code, we brought the hardware to life, using basic programming concepts like variables, if-else conditions, and input-output handling.

"The formula (duration divided by 2) divided by 29.1" helped convert the time into a measurable distance, allowing the system to trigger the buzzer and LED.

Today, you've learned important skills like working with sensors, coding logic, and circuit building. These skills are the foundation for creating innovations in robotics, automation, and security.

Task:

Now, as a task, try to create a security system using two LEDs and a buzzer.

- Design the system so one LED lights up at a medium distance, while the second LED and the buzzer activate when the object is very close.

Keep experimenting, learning, and building. This is just the beginning of your journey in technology. Well done, and here's to many more exciting projects ahead! Thank you!"