|  |
| --- |
| #include <millisDelay.h> |

|  |
| --- |
| millisDelay buzzerDelay; |

|  |
| --- |
| #define trigPin 8 //connect sensor trigPin to 8 #define echoPin 9 //connect echoPin to 9 |

|  |
| --- |
| pinMode(buzzer, OUTPUT); //set buzzer as Output pinMode(echoPin, INPUT); //set echopin as input; sensor receives the signal pinMode(trigPin, OUTPUT); //set trigPin as output; sensor transmitter sends signal |

.

|  |
| --- |
| long timeTaken, halfDistance; //32 bits number storage long thefirstDistance, currentDistance; boolean alarmBuzzDelay = false; //state of the buzzer delay boolean flagme = false; // State of the alarm boolean setAlarmTone = false; boolean alarmSet = false; boolean alarmActivate = false; |

|  |
| --- |
| long getDistance(){   digitalWrite(trigPin, LOW);  delayMicroseconds(2);  digitalWrite(trigPin, HIGH);  delayMicroseconds(10);   digitalWrite(trigPin, LOW); //now we turn off trigPin  timeTaken = pulseIn(echoPin, HIGH);   halfDistance = timeTaken\*0.034/2;   return halfDistance; //this returns the halfDistance value } |

|  |
| --- |
| void loop() { if (alarmSet){   alarmSet = false;  alarmActivate = true;   thefirstDistance = getDistance(); |

|  |
| --- |
| if (alarmActivate) { currentDistance = getDistance()+5;   if (currentDistance < thefirstDistance){   lcd.clear();  buzzerDelay.Start(10000);  alarmBuzzDelay = true; //set it to true  enterPassword(); //we callback this method. Should enter correct password to stop the alarm   } |

|  |
| --- |
| void enterPassword() { //user inputs password and compares set password  flagme = true;  ..  ..  while(flagme) { //this does not run until the buzzerDelay completes 10 second  if (alarmBuzzDelay && buzzerDelay.justFinished()) { alarmBuzzDelay = false; tone(buzzer, alarmTone); //the alarm will continue buzzing while user has to enter correct password  }  pressedKey = myKeypad.getKey(); |

|  |
| --- |
| unsigned int alarmTone=900; |

|  |
| --- |
| #define buzzer 7 //connect buzzer to pin 7 in arduino board |

|  |
| --- |
| if (pressedKey == '\*' && (tempAlarmTone.length() != 0)) { alarmTone = tempAlarmTone.toInt();  if (alarmTone > 20 && alarmTone <= 20000){  lcd.clear();  lcd.setCursor(0,0);  lcd.print("Set Tone..done!"); tone(buzzer,alarmTone,3000); //this produces the sample alarm tone set by the user for 3 seconds  setAlarmTone = false; //this will help exit the loop  } |

|  |
| --- |
| alarmActivate = false; noTone(buzzer); |