

QUICK ASSISTANCE DEVICE FOR THE ELDERLY

OBJECTIVE

To create an easy to use single button device to for senior citizens during emergencies to facilitate easy contact of family members and their hospital.

ABSTRACT

Our project uses an Atmel ATSAMW25 SoC (System on Chip) which is wifi enabled by it's WINC1500 2.4GHz IEEE® 802.11 b/g/n Wi-Fi module. The SAMD21 Cortex-M0+ 32bit low power ARM MCU is connected to Iot service like Temboo(used to connect cloud services to the microcontroller) and Amazon Web Services(the cloud service) where a notification is sent to the user's relatives and to the hospital with a press of a button and sounds a buzzer.

INTRODUCTION

Senior citizens and very elderly people depend on other people as they get old and even for day to day task they need help and they may have some existing health conditions but we cannot always be with them, there may be moments they will be alone and some emergency might happen. They will be helpless in those situations and not physically able to use a phone or any other mode of communication. So they will need immediate assistance during emergencies like heart attacks. Our device will help them contact the hospital or their loved ones with a press of a single button.

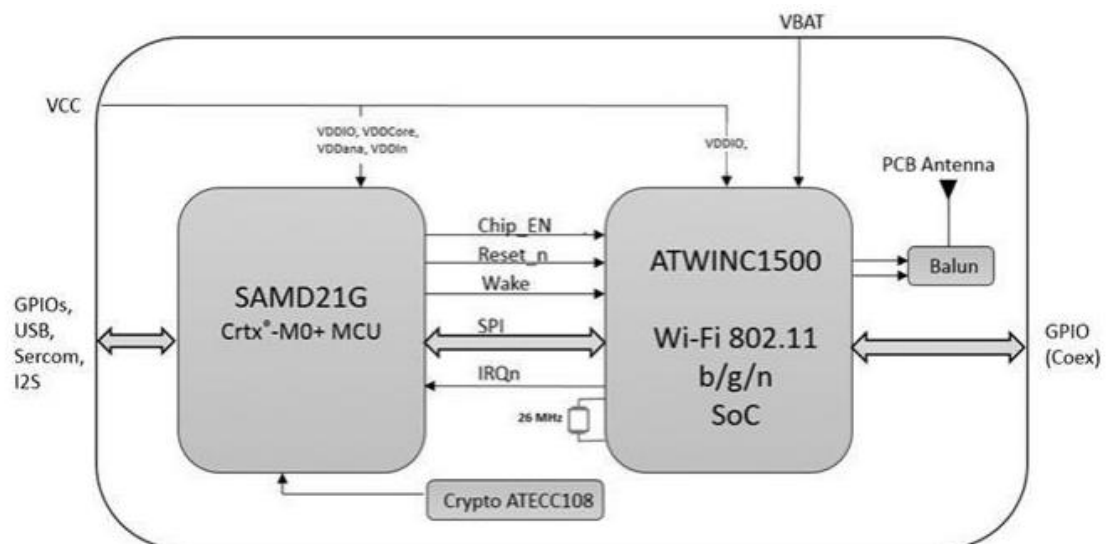
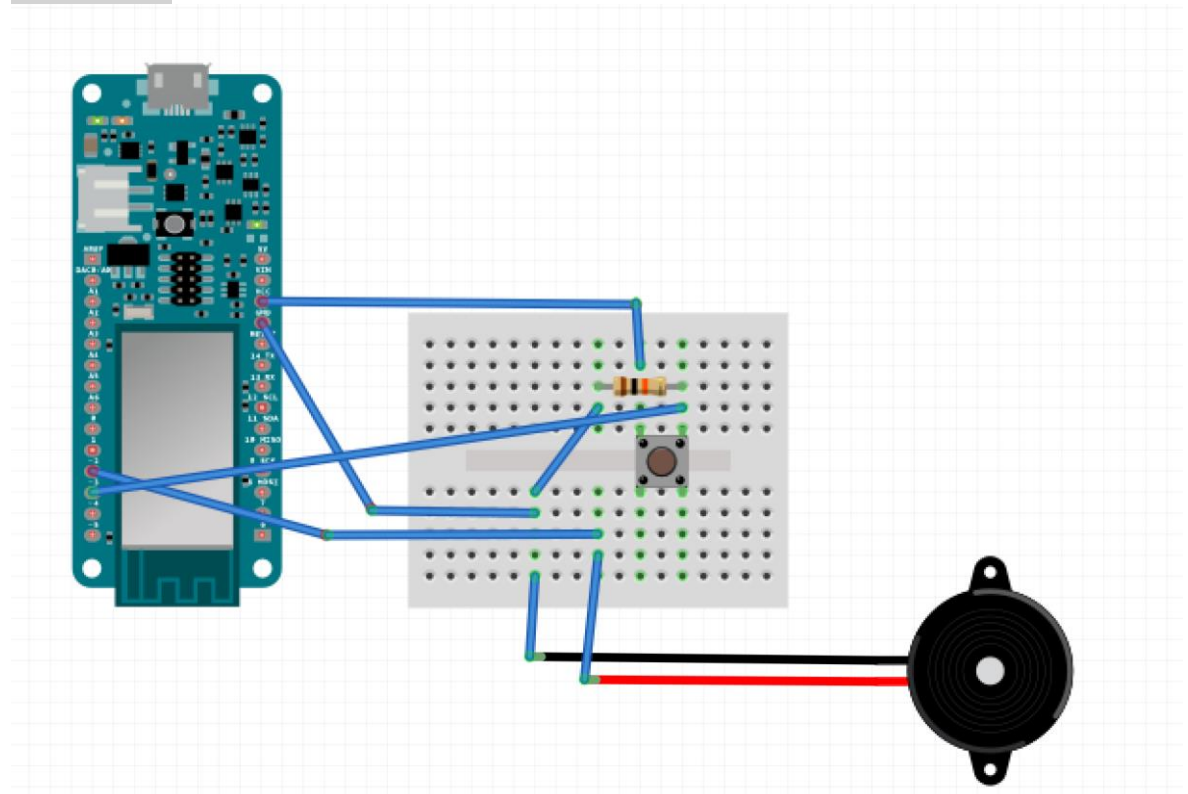
HARDWARE REQUIREMENT/DESCRIPTION

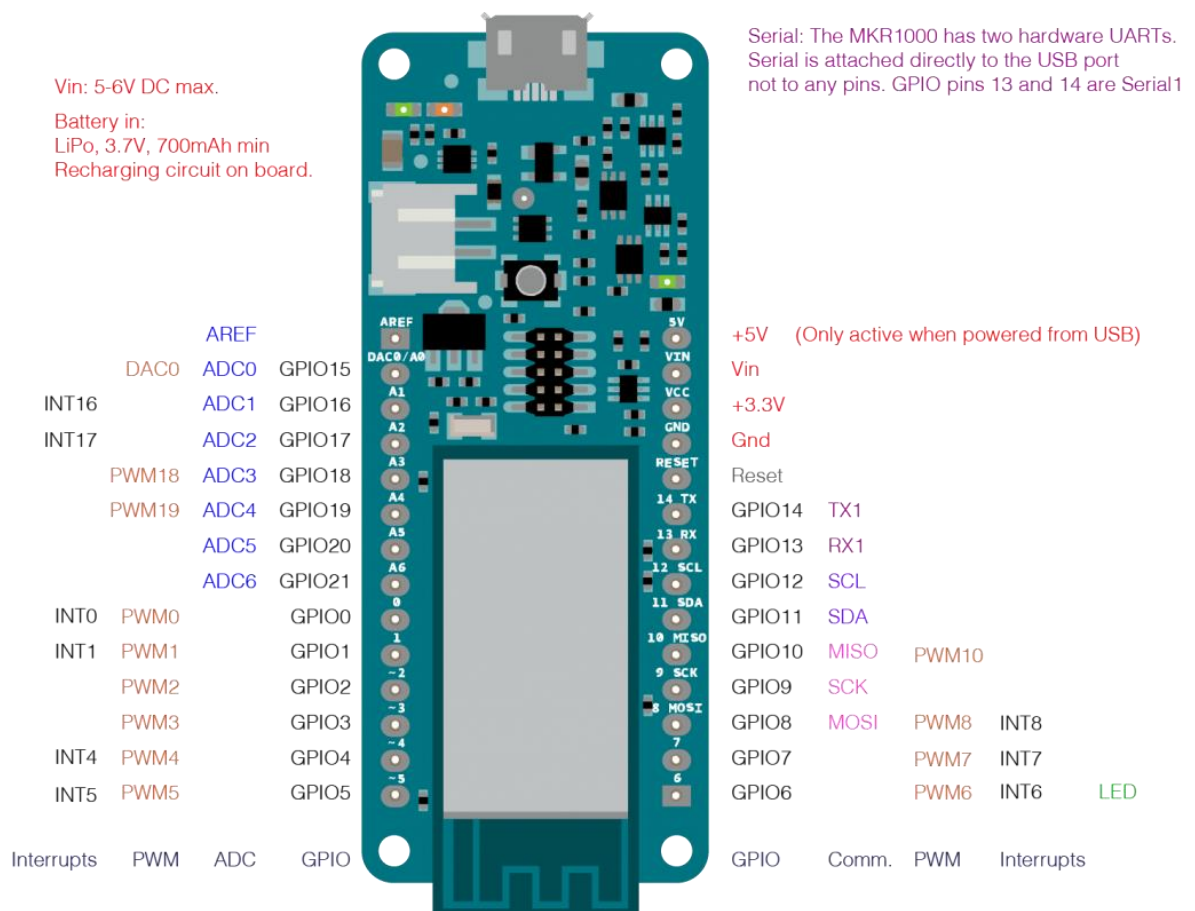
- ATSAMW25 SoC with SAMD21 Cortex-M0+ 32 bit low power ARM MCU and WINC1500 2.4GHz IEEE® 802.11 b/g/n Wi-Fi
- Pushbutton
- Buzzer
- 3.7v 1100mAh Lithium polymer battery

CIRCUIT/COMPONENT SPECIFICATIONS

Board Power Supply (USB/VIN)	5 V
Circuit Operating Voltage	3.3 V
DC Current per I/O Pin	7 mA

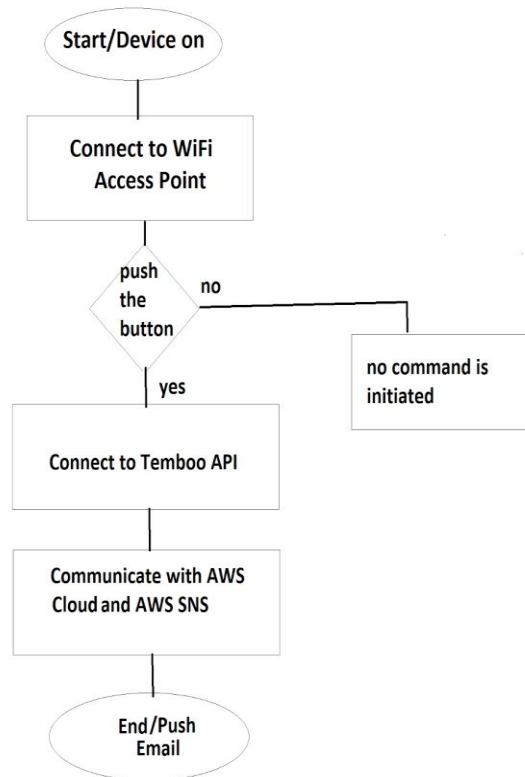
DIAGRAM



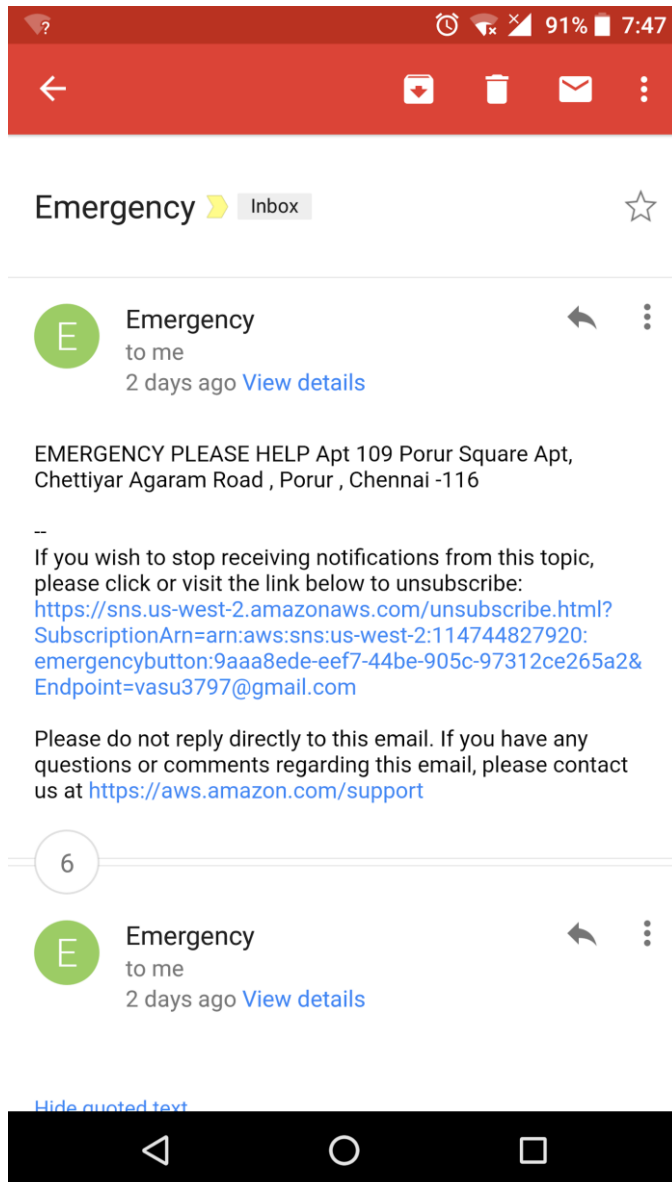


APPROACH/METHODOLOGY

This device is an internet of things based device. The ATSAMW25 SoC is connected to a pushbutton through a 10k ohm pull up resistor and a buzzer. As the pushbutton is pressed the buzzer is rings and the device through the Temboo API (which allows the microcontroller through Wifi to connect to any cloud service) accesses Amazon Web Service(the cloud service) and communicates with Amazon SNS (Simple Notification Service) and pushes an email to all the subscribed members to the service.



OUTPUT



CONCLUSIONS

Our device's first prototype has been designed and constructed. It is ready for use. We can further develop this device by using GSM instead of WiFi for connection and use GPS for dynamic location update.

REFERENCES

- [1] Arduino Organization
<https://store.arduino.cc/usa/arduino-mkr1000>
- [2] Temboo API
<https://temboo.com/library/Library/Amazon/SNS/Publish/>
- [3] Amazon Web Services

APPENDIX

MKR1000



MKR1000 has been designed to offer a practical and cost effective solution for makers seeking to add Wi-Fi connectivity to their projects with minimal previous experience in networking. It is based on the Atmel [ATSAMW25](#) SoC (System on Chip), that is part of the SmartConnect family of Atmel Wireless devices, specifically designed for IoT projects and devices.

Push Button



The pushbutton is a component that connects two points in a circuit when you press it,

Buzzer



A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.

CODE

```
#include <SPI.h>
#include <WiFi101.h>
#include <WiFiSSLClient.h>
#include <TembooSSL.h>
#include "utility/TembooGPIO.h"
#include "TembooAccount.h" // Contains Temboo account information
```

```
WiFiSSLClient client;
// We limit this so you won't use all of your Temboo calls while testing
// The number of times to trigger the action if the condition is met
```

```

int maxCalls = 10;

// The number of times this Choreo has been run so far in this sketch
int calls = 0;
//Buzzer pin
int buzzer = 1;

// Declaring sensor configs
TembooGPIOConfig tmb_buttonConfig;

// Declaring TembooSensors
TembooSensor tmb_button;

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

  // For debugging, wait until the serial console is connected
  delay(2000);

  int wifiStatus = WL_IDLE_STATUS;

  // Determine if the WiFi Shield is present
  Serial.print("\n\nShield:");
  if (WiFi.status() == WL_NO_SHIELD) {
    Serial.println("FAIL");

    // If there's no WiFi shield, stop here
    while(true);
  }

  Serial.println("OK");

  // Try to connect to the local WiFi network
  while(wifiStatus != WL_CONNECTED) {
    Serial.print("WiFi:");
    wifiStatus = WiFi.begin(WIFI_SSID, WPA_PASSWORD);

    if (wifiStatus == WL_CONNECTED) {
      Serial.println("OK");
    } else {
      Serial.println("FAIL");
    }
    delay(3000);
  }

  // Initialize sensors and configs
  tembooDigitalGPIOInit(&tmb_buttonConfig, &tmb_button, 2, LOW, INPUT);
  pinMode(buzzer, OUTPUT);
  Serial.println("Setup complete.\n");
}

```

```

void loop() {
  int sensorValue = tmb_button.read(&tmb_buttonConfig);
  Serial.println("Sensor: " + String(sensorValue));
  if (sensorValue == HIGH) {
    if (calls < maxCalls) {
      Serial.println("\nTriggered! Calling Publish Choreo...");
      tone(buzzer,500,2000);
      runPublish(sensorValue);
      calls++;
    } else {
      Serial.println("\nTriggered! Skipping to save Temboo calls. Adjust maxCalls as
required.");
    }
  }
  delay(250);
}

```

```

void runPublish(int sensorValue) {
  TembooChoreoSSL PublishChoreo(client);

  // Set Temboo account credentials
  PublishChoreo.setAccountName(TEMBOO_ACCOUNT);
  PublishChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
  PublishChoreo.setAppKey(TEMBOO_APP_KEY);
  PublishChoreo.setDeviceType(TEMBOO_DEVICE_TYPE);
  // Set Choreo inputs
  String MessageValue = "EMERGENCY PLEASE HELP Apt 109 Porur Square Apt,
Chettiyar Agaram Road , Porur , Chennai -116";
  PublishChoreo.addInput("Message", MessageValue);
  StringTopicArnValue = "arn:aws:sns:us-west-
2:114744827920:emergencybutton";
  PublishChoreo.addInput("TopicArn", TopicArnValue);
  String UserRegionValue = "us-west-2";
  PublishChoreo.addInput("UserRegion", UserRegionValue);
  String AWSAccessKeyIdValue = "AKIAIDZY3VROE6SHQFAQ";
  PublishChoreo.addInput("AWSAccessKeyId", AWSAccessKeyIdValue);
  StringAWSSecretKeyIdValue="N8c8TZ+fqneme6VmFFHyBQOp4bEKLhvCRcGZqc
PF";
  PublishChoreo.addInput("AWSSecretKeyId", AWSSecretKeyIdValue);
  String SubjectValue = "Emergency";
  PublishChoreo.addInput("Subject", SubjectValue);
  // Identify the Choreo to run
  PublishChoreo.setChoreo("/Library/Amazon/SNS/Publish");

  // Run the Choreo
  unsigned int returnCode = PublishChoreo.run();

  // Read and print the error message
  while (PublishChoreo.available()) {

```



```
    char c = PublishChoreo.read();  
    Serial.print(c);  
  }  
  Serial.println();  
  PublishChoreo.close();  
}
```