

PERFORMANCE METRICS

PROJECT TITLE :

IoT Based Safety Gadget for Child Safety Monitoring and Notification

TEAM ID : PNT2022TMID42176

TEAM LEADER : SAMEERA N

TEAM MEMBER 1 : MYTHILI T

TEAM MEMBER 2 : YOGESHWARI V

TEAM MEMBER 3 : SOWMIYA S

CIRCUITBUILD IN TINKERCARD:

We are going to build circuit for child safety gadget for monitoring and notification system using ARDUINO UNO through Tinkercad, In case of using Tinkercad we can simulate the function of the ARDUINO UNO. Majorly We use ARDUINO GEMMA, LILYPAD ARDUINO for wearable gadget, but here we are using arduino uno for this safety gadget. In this circuit, We use some of the sensors, three arduino and connection wire to build these gadget.

THE COMPONENTS:

Some of the components and sensors are used to built in the circuit, the components are

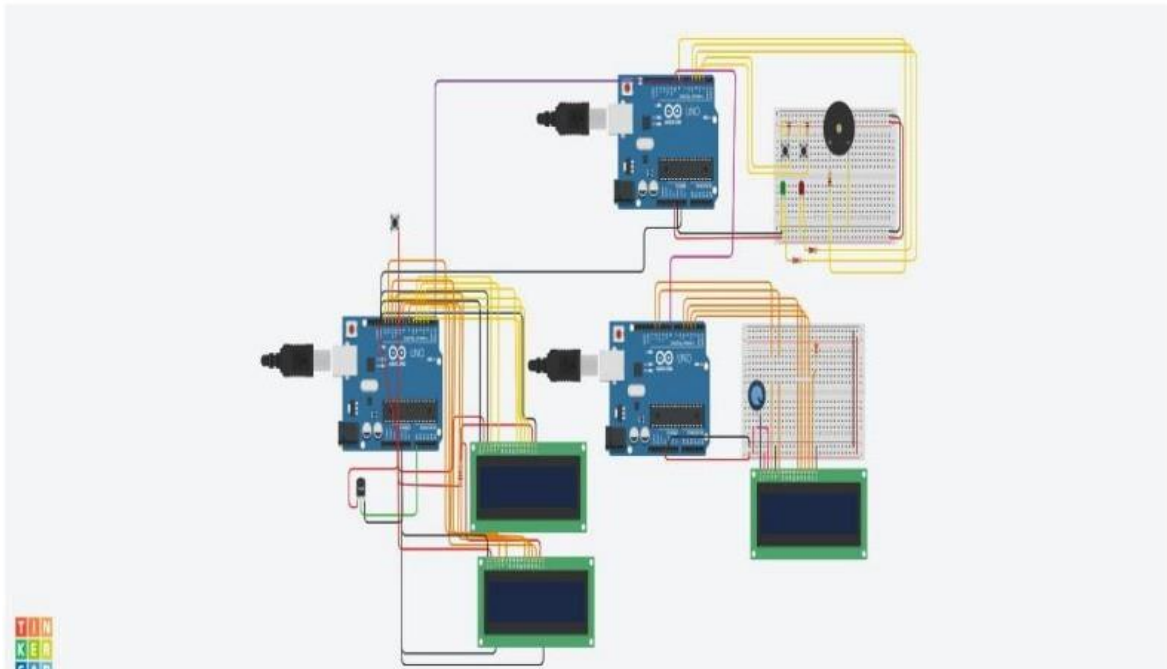
	A	B	C	D	E	F
1	Name	Quantity	Component			
2	U1, U3, U7	3	Arduino Uno R3			
3	PIEZO1	1	Piezo			
4	S1, S2, S3	3	Pushbutton			
5	R1, R2	2	10 k Ω Resistor			
6	D1	1	Green LED			
7	R3, R4, R5	3	1 k Ω Resistor			
8	D2	1	Red LED			
9	U5, U6, U4	3	LCD 16 x 2			
10	U8	1	Temperature Sensor [TMP36]			
11	R8, R6	2	220 Ω Resistor			
12	Rpot4	1	250 k Ω Potentiometer			
13						
14						

CIRCUIT BUILD-IN THROUGH TINKERCAD:

We are going to build circuit for child safety gadget for monitoring and notification system using ARDUINO UNO through Tinkercad, In case of using Tinkercad we can simulate the function of the ARDUINO UNO. Majorly We use ARDUINO GEMMA, LILYPAD ARDUINO for wearable gadget, but here we are using arduino uno for this safety gadget. In this circuit, We use some of the sensors, three arduino and connection wire to build these gadget.

THE COMPONENTS : Some of the components and sensors are used to built in the circuit, the components are

Circuit diagram:



The Working of the Circuit:

1. The first ARDUINO known as Main ARDUINO, which is connected to the buzzer and led lights. When the child is in safe status then, it notes the status as 0 by pressing through the push button and the GREEN led will blink . When the child is in danger status, the it notes the status value as 1 by pressing through the press button. And in the danger situation the RED led will blink and the buzzer would make sound like alarm.
2. The second ARDUINO is process for connection of GPS which display the location as latitude and longitude value and temperature value through LCD 12X6 display. Here this second ARDUINO is directly connected to the main ARDUINO through the digital pins.

3. And Finally the third ARDUINO, which is directly connected the main ARDUINO which is directly connected to the potentiometer and LCD 12x6 display. Which display the status of the child. These signal of the status comes from the main ARDUINO as 1 or 0. If the status value is 1, then it display the sentence as “child status” -- “ I’m safe”. If the status value is 0, “ help me”

PROGRAM:

1. Main ARDUINO program:

```
//Child safety device
```

```
char text1[] = "$GPVTG,054.7,T,034.4,M,005.5,N,010.2,K";
```

```
char text2[] =
```

```
"$GPGGA,134658.00,5106.9792,N,11402.3003,W,2,09,1.0,1048.47,M,-
```

```
16.27,M,08,AAAA*60";
```

```
int redbutton=3,greenbutton=2;
```

```
int redLED=5,greenLED=4;
```

```
int redstate=0,greenstate=0;
```

```
int buzzer=8;
```

```
int status;
```

```
int msg=9;
```

```
int greencount=0;
```

```
#define count 3
```

```
void setup()
```

```
{
```

```
pinMode(redbutton,INPUT);

pinMode(greenbutton,INPUT);

pinMode(redLED,OUTPUT);

pinMode(greenLED,OUTPUT);

attachInterrupt(digitalPinToInterrupt(redbutton),Danger,RISING);

attachInterrupt(digitalPinToInterrupt(greenbutton),Safe,RISING);

Serial.begin(9600);

pinMode(msg,OUTPUT);

digitalWrite(greenLED,HIGH);

}

//SHOW THE STATUS OF CHILD

void loop()

{

if(status)

{

tone(buzzer,100,10);

digitalWrite(msg,HIGH);

}

else

{

digitalWrite(buzzer,LOW);

digitalWrite(msg,LOW);
```

```
}  
  
//GPS  
  
delay(200);  
  
Serial.write(text1);  
  
Serial.write("/");  
  
delay(500);  
  
Serial.write(text2);  
  
delay(500);  
  
}  
  
void Danger()  
  
{  
  
digitalWrite(redLED,HIGH);  
  
digitalWrite(greenLED,LOW);  
  
status=1;  
  
}  
  
void Safe()  
  
{  
  
greencount=greencount+1;  
  
  
  
if(greencount==count)  
  
{  
  
digitalWrite(greenLED,HIGH);
```

```
digitalWrite(redLED,LOW);  
  
status=0;  
  
greencount=0;  
  
}  
  
}
```

2. GPS:

```
#include <LiquidCrystal.h>  
  
#define ADC_VREF_TYPE ((0<<REFS1) | (0<<REFS0) | (0<<ADLAR))  
  
float stepADC = 0.0048828125;  
  
LiquidCrystal lcd1(5, 4, 3, 13, 2, 6);  
  
LiquidCrystal lcd2(12, 11, 10, 9, 8, 7);  
  
char sSecventa[119];  
  
//declare global deoarece in cazul calculatiilor din ecranul 2  
  
//memoria este deja scrisa si ar fi trebuit o alta initializare  
  
//cu mai multe varibile, etc..  
  
static char *aValori[2];  
  
static char *token;  
  
static char *aValori1[9];  
  
static char *token1;  
  
static char *aValori2[15];  
  
static char *token2;  
  
int contor = 0;
```

```

bool ecran = 0;

bool stareButon = 0;

bool stareButonAnterioara = 0;

//.....

unsigned int citesteADC(unsigned int adc_input)
{
    ADMUX = adc_input | ADC_VREF_TYPE
    //delay necesar pentru stabilirea ADC pt tens
    //de intrare analogica
    delayMicroseconds(10);

    //start conversie
    ADCSRA |= (1<<ADSC);

    //Asteptare finalizare conversie
    while ((ADCSRA & (1<<ADIF))==0){}

    ADCSRA |= (1<<ADIF);

    //returnare rezultat pe 16 bitia
    return ADCW;
}

//cod recunoastere/segmentare secventa
void segmentareSecventa()

```



```
{  
    //segmentare secventa principala  
    token = strtok(sSecventa, "/");  
    static int increment = 0;  
    while (token != NULL)  
    {  
        aValori[increment++] = token;  
  
        token = strtok(NULL, "/");  
    }  
  
    //segmentare secventa 1  
    token1 = strtok(aValori[0], ",");  
    static int increment1 = 0;  
    while (token1 != NULL)  
    {  
        aValori1[increment1++] = token1;  
  
        token1 = strtok(NULL, ",");  
    }  
  
    //segmentare secventa 2  
    token2 = strtok(aValori[1], ",");
```

```

static int increment2 = 0;

while (token2 != NULL)
{
    aValori2[increment2++] = token2;

    token2 = strtok(NULL, ",");
}

//incheiere segmentare secvente -----
}

void ecranulUnu()
{
    //afisare viteza

    static float iViteza = 0;

    if (strcmp(aValori1[0], "$GPVTG") == 0)
    {
        iViteza = atof(aValori1[7]);
    }

    //setare cursor: coloana 0, linia 1

    lcd1.setCursor(0, 1);

    lcd1.print("Vit: ");

    lcd1.print(iViteza);

    lcd1.print("km/h");
}

```

```
//afisare timp
static long int temp[4];

static int iSateliti = 0;

if (strcmp(aValori2[0], "$GPGGA") == 0)
{
    //stocare sir
    temp[0] = atof(aValori2[1]);

    //stocare secunde
    temp[3] = temp[0] % 100;
    temp[0] = temp[0] / 100;

    //stocare minute
    temp[2] = temp[0] % 100;
    temp[0] = temp[0] / 100;

    //stocare ore
    temp[1] = temp[0];

    iSateliti = atoi(aValori2[7]);
}

//setare cursor: coloana 0, linia 0
lcd1.setCursor(0, 0);
```

```
lcd1.print("Timp: ");
```

```
lcd1.print(temp[1]);
```

```
lcd1.print(":");
```

```
lcd1.print(temp[2]);
```

```
lcd1.print(":");
```

```
lcd1.print(temp[3]);
```

```
//setare cursor: coloana 0, linia 3
```

```
lcd2.setCursor(0,0);
```

```
lcd2.print("Nr. sat: ");
```

```
lcd2.print(iSateliti);
```

```
//citire termometru de pe pinul analogic "0"
```

```
unsigned int sensorValue = citesteADC(0);
```

```
//calculare temperatura in functie de valorile din datasheetul termometrului
```

```
float fTemperature = (stepADC*sensorValue-0.5)*100;
```

```
//setare cursor: coloana 0, linia 3
```

```
lcd2.setCursor(0, 1);
```

```
lcd2.print("Temp: ");
```

```
lcd2.print(fTemperature, 2);
```

```

lcd2.print(" C");

//incheiere prelucrare ecran unu-----
}

//afisare/selectare date pe ecranul doi
void ecranulDoi()
{
    //segmentareSecventa();

    //calculare, afisare latitudine
    static float fLatitudine = 0;
    static float fLongitudine = 0;
    static float fAltitudine = 0;
    if (strcmp(aValori2[0], "$GPGGA") == 0)
    {
        fLatitudine = atof(aValori2[2]);
        fLongitudine = atof(aValori2[4]);
        fAltitudine = atof(aValori2[9]) - atof(aValori2[11]);
    }

    //afisare date

    //setare cursor: coloana 0, linia 1
    lcd1.setCursor(0, 1);

    lcd1.print("Lat: ");

    lcd1.print((int)fLatitudine/100+((int)(fLatitudine)%100+(fLatitudine-

```

```

(int)(fLatitudine)))/60);

lcd1.print(aValori2[3]);

//setare cursor: coloana 0, linia 2

lcd2.setCursor(0, 0);

lcd2.print("Long: ");

lcd2.print((int)fLongitudine/100+((int)(fLongitudine)%100+(fLongitudine-
(int)(fLongitudine)))/60);

lcd2.print(aValori2[5]);

//setare cursor: coloana 0, linia 3

lcd2.setCursor(0, 1);

lcd2.print("Alt: ");

lcd2.print(fAltitudine);

lcd2.print("m");

//incheiere prelucrare ecran doi-----
}

//.....

//.....SETUP.....

//.....

void setup()

{

Serial.begin(9600);

```

```

lcd1.begin(2,16);

lcd2.begin(2,16);

}

//.....

//.....LOOP.....

//.....

void loop()

{

if(contor < 1)

{

Serial.readBytes(sSecventa, 119);

Serial.println(sSecventa);

segmentareSecventa();

contor++;

}


//citire stare buton

/*stareButon = (PIND & (1 << PIND2));

if(stareButon != stareButonAnterioara)

{

delay(200);

```

```
ecran = !ecran;
```

```
if( stareButon == 0)
```

```
{
```

```
stareButon = !stareButon;
```

```
}
```

```
}
```

```
stareButonAnterioara = stareButon;
```

```
*/
```

```
ecranulUnu();
```

```
delay(1000);
```

```
lcd1.clear();
```

```
lcd2.clear();
```

```
ecranulDoi();
```

```
delay(1000);
```

```
lcd1.clear();
```

```
lcd2.clear();
```

```
}
```

```
3.GSM module
```

```
//parents and friends screen
```



```
#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

int status=8;

int data;

void setup()

{

  pinMode(status,INPUT);

  lcd.begin(16, 2);

  lcd.setCursor(0,0);

  lcd.print("Child status");

  pinMode(8,INPUT);

  Serial.begin(9600);

}

void loop()

{

  data=digitalRead(status);

  Serial.println(data);

  if(data==HIGH)

  {

    lcd.setCursor(0,1);

    lcd.print("Help me!");

  }

}
```

```
if(data==LOW)
{
    lcd.setCursor(0,1);
    lcd.print("I'm safe!");
}
}
```

```
//delay necesar pentru stabilirea ADC pt tens
```

```
//de intrare analogica
```

```
delayMicroseconds(10);
```

```
//start conversie
```

```
ADCSRA |= (1<<ADSC);
```

```
//Asteptare finalizare conversie
```

```
while ((ADCSRA & (1<<ADIF))==0){}
```

```
ADCSRA |= (1<<ADIF);
```

```
//returnare rezultat pe 16 bitia
```

```
return ADCW;
}

//cod recunoastere/segmentare secventa
void segmentareSecventa()
{
    //segmentare secventa principala
    token = strtok(sSecventa, "/");
    static int increment = 0;
    while (token != NULL)
    {
        aValori[increment++] = token;

        token = strtok(NULL, "/");
    }

    //segmentare secventa 1
    token1 = strtok(aValori[0], ",");
    static int increment1 = 0;
    while (token1 != NULL)
    {
```

```
aValori1[increment1++] = token1;
```

```
token1 = strtok(NULL, ",");
```

```
}
```

```
//segmentare secventa 2
```

```
token2 = strtok(aValori[1], ",");
```

```
static int increment2 = 0;
```

```
while (token2 != NULL)
```

```
{
```

```
aValori2[increment2++] = token2;
```

```
token2 = strtok(NULL, ",");
```

```
}
```

```
//incheiere segmentare secvente -----
```

```
}
```

```
void ecranulUnu()
```

```
{
```

```
//afisare viteza
```

```
static float iViteza = 0;
```

```
if (strcmp(aValori1[0], "$GPVTG") == 0)
```

```
{  
iViteza = atof(aValori1[7]);  
}  
  
//setare cursor: coloana 0, linia 1  
lcd1.setCursor(0, 1);  
lcd1.print("Vit: ");  
lcd1.print(iViteza);  
lcd1.print("km/h");  
  
//afisare timp  
static long int temp[4];  
static int iSateliti = 0;  
if (strcmp(aValori2[0], "$GPGGA") == 0)  
{  
//stocare sir  
temp[0] = atof(aValori2[1]);  
//stocare secunde  
temp[3] = temp[0] % 100;  
temp[0] = temp[0] / 100;  
//stocare minute
```

```
temp[2] = temp[0] % 100;
```

```
temp[0] = temp[0] / 100;
```

```
//stocare ore
```

```
temp[1] = temp[0];
```

```
iSateliti = atoi(aValori2[7]);
```

```
}
```

```
//setare cursor: coloana 0, linia 0
```

```
lcd1.setCursor(0, 0);
```

```
lcd1.print("Timp: ");
```

```
lcd1.print(temp[1]);
```

```
lcd1.print(":");
```

```
lcd1.print(temp[2]);
```

```
lcd1.print(":");
```

```
lcd1.print(temp[3]);
```

```
//setare cursor: coloana 0, linia 3
```

```
lcd2.setCursor(0,0);
```

```
lcd2.print("Nr. sat: ");
```

```
lcd2.print(iSateliti);
```

```
//citire termometru de pe pinul analogic "0"
```

```
unsigned int sensorValue = citesteADC(0);
```

```
//calculare temperatura in functie de valorile din datasheetul  
termometrului
```

```
float fTemperature = (stepADC*sensorValue-0.5)*100;
```

```
//setare cursor: coloana 0, linia 3
```

```
lcd2.setCursor(0, 1);
```

```
lcd2.print("Temp: ");
```

```
lcd2.print(fTemperature, 2);
```

```
lcd2.print(" C");
```

```
//incheiere prelucrare ecran unu-----
```

```
}
```

```
//afisare/selectare date pe ecranul doi
```

```
void ecranulDoi()
```

```
{
```

```
//segmentareSecventa();
```

```
//calculare, afisare latitudine

static float fLatitudine = 0;

static float fLongitudine = 0;

static float fAltitudine = 0;

if (strcmp(aValori2[0], "$GPGGA") == 0)

{

fLatitudine = atof(aValori2[2]);

fLongitudine = atof(aValori2[4]);

fAltitudine = atof(aValori2[9]) - atof(aValori2[11]);

}

//afisare date

//setare cursor: coloana 0, linia 1

lcd1.setCursor(0, 1);

lcd1.print("Lat: ");

lcd1.print(((int)fLatitudine/100+((int)(fLatitudine)%100+(fLatitudine-
(int)(fLatitudine)))/60);

lcd1.print(aValori2[3]);

//setare cursor: coloana 0, linia 2

lcd2.setCursor(0, 0);

lcd2.print("Long: ");
```



```

lcd2.print((int)fLongitudine/100+((int)(fLongitudine)%100+(fLongitudin
e-
(int)(fLongitudine)))/60);

lcd2.print(aValori2[5]);

//setare cursor: coloana 0, linia 3

lcd2.setCursor(0, 1);

lcd2.print("Alt: ");

lcd2.print(fAltitudine);

lcd2.print("m");

//incheiere prelucrare ecran doi-----
}

//_____

//____SETUP____

//_____

void setup()

{

Serial.begin(9600);

lcd1.begin(2,16);

```

```
lcd2.begin(2,16);  
}  
  
//_____  
  
//____LOOP_____  
  
//_____  
  
void loop()  
{  
  if(contor < 1)  
  {  
    Serial.readBytes(sSecventa, 119);  
    Serial.println(sSecventa);  
    segmentareSecventa();  
    contor++;  
  }  
  
  //citire stare buton  
  
  /*stareButon = (PIND & (1 << PIND2));  
  
  if(stareButon != stareButonAnterioara)  
  {
```

```
delay(200);
```

```
ecran = !ecran;
```

```
if( stareButon == 0)
```

```
{
```

```
stareButon = !stareButon;
```

```
}
```

```
}
```

```
stareButonAnterioara = stareButon;
```

```
*/
```

```
ecranulUnu();
```

```
delay(1000);
```

```
lcd1.clear();
```

```
lcd2.clear();
```

```
ecranulDoi();
```

```
delay(1000);
```

```
lcd1.clear();
```

```
lcd2.clear();
```

```
}
```

3.GSM module

```
//parents and friends screen
```

```
#include <LiquidCrystal.h>
```

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

```
int status=8;
```

```
int data;
```

```
void setup()
```

```
{
```

```
  pinMode(status,INPUT);
```

```
  lcd.begin(16, 2);
```

```
  lcd.setCursor(0,0);
```

```
  lcd.print("Child status");
```

```
  pinMode(8,INPUT);
```

```
  Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
  data=digitalRead(status);
```

```
  Serial.println(data);
```

```
if(data==HIGH)
{
  lcd.setCursor(0,1);
  lcd.print("Help me!");
}
if(data==LOW)
{
  lcd.setCursor(0,1);
  lcd.print("I'm safe!");
}
}
```