ChildCarMonitor

Nik Grebovšek

63200102

## Opis problema

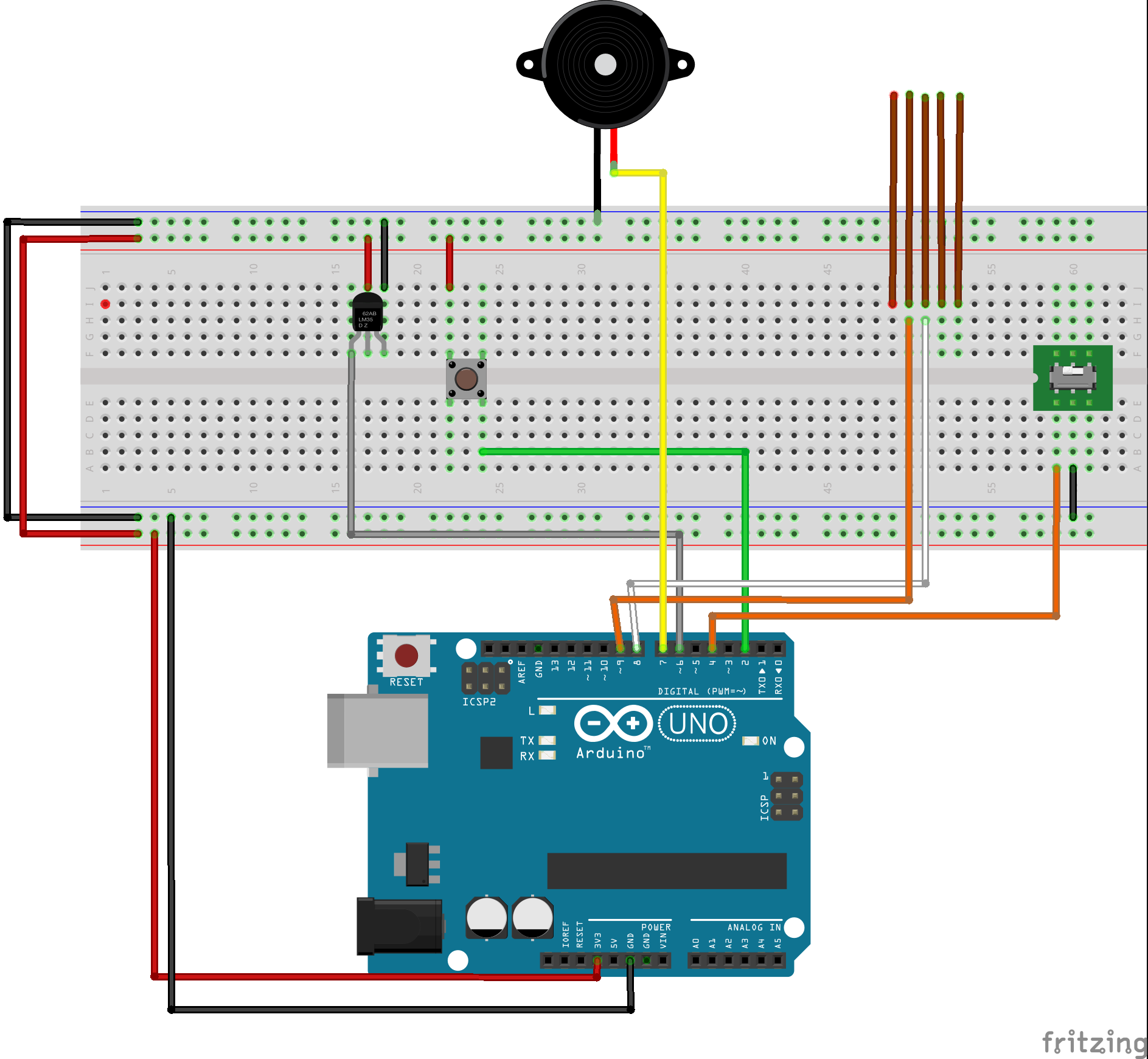
Na žalost se prevečkrat zgodi, da ljudje v svojem avtu pozabijo otroka ali svojega domačega ljubljenčka. Ko temperatura v avtu zraste previsoko, je lahko za otroka/žival to smrtno nevarno.

## Rešitev problema

Vezje, ki konstantno preverja, ali je temperatura v avtu previsoka in ali je, o tem obvesti lastnika avta.

## Opis vezja

Za vezje sem uporabil Arduino, merilec temperature in GPS modul. Vsi skupaj so povezani na Arduino. Diagram vezja je spodaj.



(Za merjenje temperature sem uporabil drugačen tip senzorja. Rjave žice so povezane na GPS modul. Buzzer na moji plošči ima tri pine, dva sta za napajanje, tretji je pa signal.)

## Opis delovanja

Če je dvostopenjski gumb na položaju desno, to pomeni, da je na zadnjem sedežu nekaj težjega (otrok/žival). V tem primeru Arduino na vsak TimerDelay preveri stanje naprav. Najprej preveri, če je bila temperatura izmerjena več kot je TempertureAlarm. In če je, pokliče metodo alarm, ki poskrbi za obveščanje uporabnika.

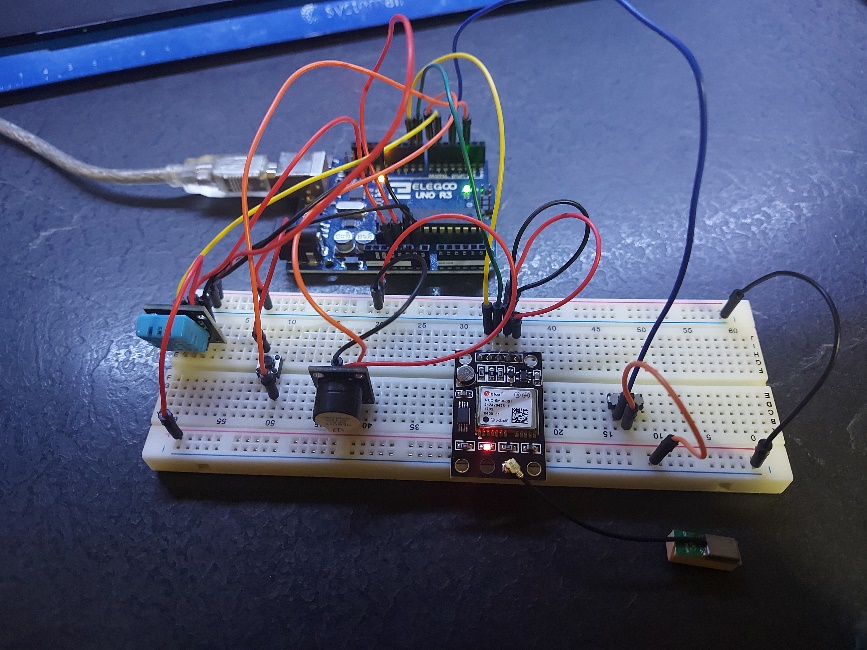
Potem z rekurzivno funkcijo pridobi temperaturo, ki jo uporabi naprej. (Rekurzivna je zato, saj se včasih zgodi, da ne more prebrati temperature in se zato izvede večkrat.)

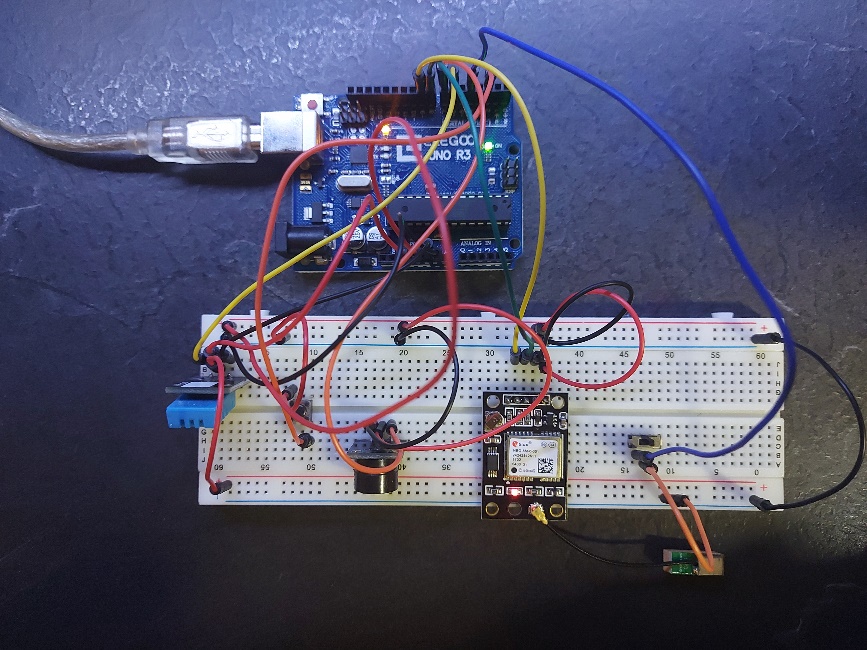
Nato glede na temperaturo naredi naslednje: če je večja od WarningTemp potem warningTempCounter, če pa je manjša, pa pomanjša.

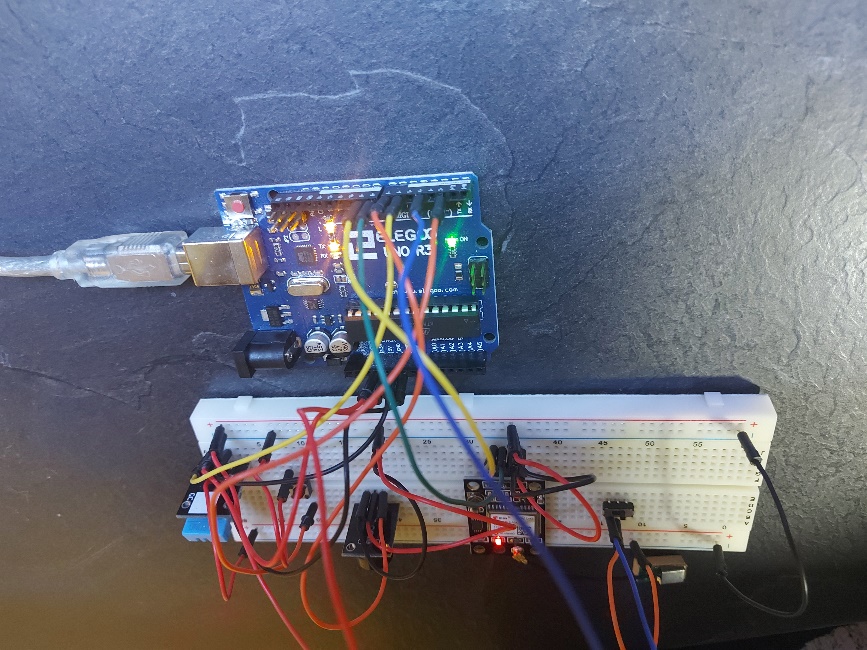
Ko se kliče funkcija alarm, se vklopi piskač, ki opozarja mimoidoče. Prav tako pa odpre povezavo z GPS modulom, od katerega v nekaj poskusih poskusi pridobiti lokacijo. Le ta metoda zgenerira sporočilo, ki bi se poslalo na lastnikov telefon.

Ker mi je zmanjkalo časa in ker sem imel težave z WIFI modulom, mi tega ni uspelo narediti. Želel pa sem uporabiti svoj [SMSsenderAPI](https://nikigre.si/sl/sms-sender-api/).

## Slike







## Koda

#include "DHT.h"

#include <TinyGPSPlus.h>

#include <SoftwareSerial.h>

#define DEBUG true

//Variables for setting up the child car monitor

#if DEBUG == true

#define TimerDelay 1000 \* 2 // check sensors at X seconds

#define WarningTemp 25 //25 °C is too much

#define TempertureAlarm 10 //how many times TimerDelay must pass that we signal alarm

#define MaxWarningTempCounter 2 \* TempertureAlarm //MAX temp counter

#else

#define TimerDelay 1000 \* 10 // check sensors at X seconds

#define WarningTemp 35 //35 °C is too much

#define TempertureAlarm 30 //how many times TimerDelay must pass that we signal alarm

#define MaxWarningTempCounter 2 \* TempertureAlarm //MAX temp counter

#endif

//GPS module parameters

#define gpsRXpin 8

#define gpsTXpin 9

#define gpsBaud 9600

//Temperature sensor parameters

#define DHTPIN 6

#define DHTTYPE DHT11

//Buzzer parameters

#define BuzzerPin 7

//Reset button parameters

#define ResetButton 2

//Weight sensor

#define WeightSensorPin 4

DHT dht(DHTPIN, DHTTYPE);

TinyGPSPlus gps;

SoftwareSerial ss(gpsRXpin, gpsTXpin);

char buff[100]; //For sprintf method

char temperatureString[6]; //For displaying temperature as string

int timer; //Timer that counts loop()

int warningTempCounter = 0; //How many times the temperature was above limit

bool smsSent = false;

void setup() {

Serial.begin(9600);

ss.begin(gpsBaud);

pinMode(BuzzerPin, OUTPUT);

pinMode(ResetButton, INPUT\_PULLUP);

pinMode(WeightSensorPin, INPUT\_PULLUP);

attachInterrupt(digitalPinToInterrupt(ResetButton), resetData, CHANGE); //We make an interrupt on pin so we can restart all parameters on click

dht.begin();

}

void loop() {

if(digitalRead(WeightSensorPin)==LOW)

return;

sprintf(buff, "%d\twarningTempCounter > TempertureAlarm \t %d > %d", timer, warningTempCounter, TempertureAlarm);

Serial.println(buff);

if(warningTempCounter > TempertureAlarm) { //Check if we need to produse an alarm

alarm();

} else {

digitalWrite(BuzzerPin, LOW);

}

float temperature = getTemperature(0);

sprintf(buff, "%d\ttemperature >= WarningTemp \t %s >= %d", timer, temperatureString, WarningTemp);

Serial.println(buff);

if(temperature >= WarningTemp) { //Check if temperature is above alowed limit

if(warningTempCounter != MaxWarningTempCounter)

warningTempCounter++;

} else {

if(warningTempCounter > 0)

warningTempCounter--;

}

//Delay checking

delay(TimerDelay);

timer++;

Serial.println(" ");

}

void alarm() {

if(smsSent == false)

{

smsSent = true;

String link = getGPSstring();

sprintf(buff, "Pozdravljeni!\nV vašem avtu je %s stopinj in v avtu je zaznan otrok/žival. Lokacija avta:", temperatureString);

Serial.print(buff);

Serial.println(link);

}

sprintf(buff, "%d\tALARM!!!", timer);

Serial.println(buff);

digitalWrite(BuzzerPin, HIGH);

}

float getTemperature(int q) {

if(q > 20)

{

Serial.println(F("Failed to read from DHT sensor 20 tries!"));

return -1;

}

float temperature = dht.readTemperature();

if(isnan(temperature)) {

delay(random(100, 200));

return getTemperature(q+1);

} else {

dtostrf(temperature, 4, 2, temperatureString);

sprintf(buff, "%d\t %s °C", timer, temperatureString);

Serial.println(buff);

}

return temperature;

}

void resetData()

{

warningTempCounter=0;

digitalWrite(BuzzerPin, LOW);

smsSent = false;

}

String getGPSstring()

{

int tries = 0;

do

{

if(++tries > 20)

break;

while (ss.available())

gps.encode(ss.read());

Serial.print("Trying to get GPS data!");

Serial.println(tries);

delay(random(1000, 2000));

} while(!gps.location.isValid());

if(!gps.location.isValid())

{

return "Ni GPS signala!";

}

else

{

char data[20];

String rez = "https://l.nikigre.si/s.php?l=";

dtostrf(gps.location.lat(), 20, 10, data);

rez = rez + String(data);

rez = rez + ",";

dtostrf(gps.location.lng(), 20, 10, data);

rez = rez + String(data);

return rez;

}

}