```
// <<<<<<<<<<<<   EPIDINO   >>>>>>>>>>>>>

char tel1[20] = "############";     // Nachtdienst
byte tel = 2; // 1: Nur tel1 waehlen   2: nach tel1 auch tel2 waehlen
char tel2[20] = "############";  // EpiHandy
char tel3[20] = "############";  // Nachtdienst
char tel4[20] = "############";  // Handy Eltern

String SmsText = "Epidino";

//  ++++  Version: Zur Anzeige auf dem Monitor
String verNo = "Epidino_V5_180121_1902";

//  ++++  Target3001! File: Epidino-V5 180120(2203).T3001

INFO: AVR ISP

/*
< ! > : Parameters to be changed [Mit < ! > sind die Stellen markiert, die geaendert werden koennen]

Connection for piezo sensor or Accelerometer [Anschlussmoeglichkeit von PiezoSensor und / oder Beschleunigungssensor ]
  For Accelerometer sensor you need 5V supply to the sensor unit.
  [ BeiVerwendung enes Beschleunigungssensors wir 5V Versorgung zur Snsoreinheit benoetigt.

    Momentan werden nur parallel geschaltete Piezo - Elemente verwendet, da dieser Sensor einwandfrei arbeitet.

    If no Sensor is connected, a plug with connection to ground is used.

    Duration: Waiting time for the next sensor activity. [ Wartezeit fuer die naechste zu registrierende Sensoraktivitaet.]
    AlarmShreshold: After how many seconds sensor activity an alarm will be triggered. [ Nach wieviel Sekunden Aktivitaet der Alarm ausgeloest
wird. ]

    <<< GSM Shield and Relay output >>>
    <<< GSM Shield >>>
    Advantage: Easy to use. You can program it as you like. For example to send a SMS again if
    stop button was not pressed.
    Disadvantage: The phone number must be in the code and can not be changed by others.
    But you can add 3 phone numbers which can be selected by Pot4.

    [Vorteil: Kompakt, einfach aufstecken und schon fertig. Es kann selbst so programmiert werden, wie
     man es wuenscht. Zum Beispiel nach einer gewissen Zeit noch einmal waehlen lassen, falls niemand gekommen
     ist (Stopp - Taste wurde nicht gedrueckt)
     Es kann ein Handy oder auch ein Festnetz - Telefon angerufen werden.
     Nachteil: Soll ein anderes Telefon / Handy angerufen werden, so muss die Nummer hier im Code geaendert und
     neu geflashed werden.
     Es koennen aber 3 Telefonnummern programmiert werden, die mit Pot4 gewaehlt werden koennen.]

    <<< Relay output>>>
    Advantage: You can use any GSM dialler or just a buzzer which is connected via a cable to another room.
    [ Vorteil: Es kann ein beliebiges GSM - Wahlgeraet verwendet oder auch nur eine einfache Klingel angeschlossen
      werden.

      Ein Epi - Care Geraet kann angeschlossen werden (z.B. Epi - Care free). Da bei Einschalten dieses Geraetes am Ausgang ein Impuls
      auftritt, muss verhindert werden, dass dieser einen Alarm ausloest (bei Einschalten des Epidinos und auch bei Stoppen
          eines Fehlalarms). Hierfuer ist die "ResetPause": Nach Aus - und Wiedereinschalten des Relais (bei Druecken der Stopptaste)
      wird bis nach dem Ausgangsimpuls gewartet.
      * /

#include <GSM.h>
#define PINNUMBER ""
      // Initialisation of the library [Initialsieren der Library]
      GSM gsmAccess;
      GSM_SMS sms;

      // ---- GSM  --------------------------------------
      byte GsmYn = 1;  // 1:GSM Shield is used   0:GSM Shield is not used [ 1:Mit GSM Shield    0:Ohne GSM Shield ]
      // GSM Y/N socket:  If Jumper inserted, then GsmYn=1   If no Jumper: GsmYn=0

      int AlarmDelay = 20000;         // <!> Wait until the alarm is triggered [Wartezeit bis der Alarm ausgeloest wird]
      int RelayOnDuration = 10000;    // <!> How long the alarm relay should remain ON [Wie lange das AlarmRelais anbleiben soll]
      byte RelayActive = 0;           // Whether Relay is active or not [Ob Relay aktiv ist oder nicht] (0=Nicht aktiv  1=Aktiv)
      byte GsmActive = 0;             // 0:GSM shield is not active, 1:GSM shield is active [0:GSM ist nicht aktiv  1:aktiv]
      int ResetPause = 4000;          //2000 <!> Wait after ResetRelay has resetted the Epi-Care unit to supress output signal of Epi-Care
      //                                    [Wartezeit nach Zurueckssetzen des Epi-Care]

      // For monitoring 0=OFF 1=ON [ Monitor-Ausgabe waehlen (nur fuer Programmierungszwecke)]
```

```
    byte mon1 = 1;

    // <<<<< Digital In and Outputs >>>>>
    //byte GsmRxPin = 2;       // P4 Pin2 ist fuer GSM festgelegt
    //byte GsmTxPin = 3;       // P5 Pin3 ist fuer GSM festgelegt
    //byte ModemResetPin = 7;  // P13 GSM Shield Reset
    byte SosPin = 4;           // P6 for connection of a SOS button which can be placed anywhere in the room
    byte GsmYnPin = 5;         // P11 for selecting GSM Shield
    byte AlarmRelayPin = 6;    // P12 AlarmRelay
    byte ModemResetPin = 7;    // P13 GSM Shield Reset
    byte AlarmINpin = 8;       // P14 Alarm Epi-Care1
    byte ResetRelayPin = 9;    // P15 Relay for resetting Epi-Care units [Relay zum Zuruecksetzen der beiden Epi-Care]
    byte SpeakerPin = 10;      // P16 Lautsprecher (Buzzer)
    byte StopButtonPin = 11;   // P17 Stopp-Taste
    byte RedLedPin = 12;       // P18 shows sensor activity, lights up when alarm is triggered [zeigt Sensoraktivitaet an, leuchtet stetig bei
Alarm]
    byte GreenLedPin = 13;     // P19 lights up at first sensor activity [Leuchtet bei der ersten Sensoraktivitaet auf]

    // <<<<< Analog Inputs >>>>>
    int Pot1Pin = A1; // P25 [Pot2] <TimerAlarm>
    int Pot2Pin = A2; // P26 [Pot3] <Duration>
    int Pot3Pin = A3; // P26 [Pot3] <SensThreshold> Sensitivity of Sensor
    int Pot4Pin = A0; // P23 [Pot4] <Tel.No. selection>

    byte Sens1Pin = A5; // P28 // Sensor connection
    byte Sens2Pin = A4; // P27 // Sensor connection

    byte ST = 0;     //  0:Initial status   2:Sensor activated 1:Red/Green-LED reset   9:Alarm
    byte LD = 0;     //  0:RedLed=OFF  GreenLed=OFF  1 to 5:RedLed=ON  GreenLed=OFF   6...: RedLed=OFF  GreenLed=ON
    int BZ = 0;      //  for speaker control


    unsigned long TimerStart;      // Timer start time [Startzeit des Timers] TimerStart
    float Duration;               // [Pot2] Waiting phase during monitoring period [ Wartephase innerhalb der Ueberwachung ]
    unsigned long TimerEnd;        // TimerStart + Duration [Ende der Wartezeit fuer die naechste Aktivitaet waehrend der
Aktivitaets-Registrierdauer]
    unsigned long AlarmShreshold;  // [Pot1] AlarmShreshold [ Alarmschwelle ]
    unsigned long TimerAlarm;      // TimerAlarm = TimerStart + AlarmThreshold

    int Sens1Val = 0;          // sensor values [eingelesene Sensorwerte]
    int Sens1Threshold = 350;  // [Pot3] determins at which threshold the counter should be increased

    int vvv = 0;  // general variable [allgemeine Variable]
    int www = 0;  // general variable [allgemeine Variable]

 void setup() {
 pinMode(GreenLedPin, OUTPUT);
 pinMode(RedLedPin, OUTPUT);
 pinMode(SpeakerPin, OUTPUT);
 // pinMode(ModemResetPin, OUTPUT);  // GSM-Reset
 pinMode(AlarmRelayPin, OUTPUT);  // Relay
 digitalWrite(AlarmRelayPin, HIGH);
 pinMode(AlarmINpin, INPUT);     // Alarm von Epi-Care
 pinMode(ResetRelayPin, OUTPUT);  // ResetRelay to switch Epi-Care off and on again [ResetRelay zum Aus- und Wiedereinschalten des Epi-Care]
 digitalWrite(ResetRelayPin, HIGH); // High, so that Epi-Care is supplied with 12V power
 pinMode(StopButtonPin, INPUT);
 pinMode(GsmYnPin, INPUT);
 pinMode(SosPin, INPUT);
 tone(SpeakerPin, 1500);
 delay(200);
 noTone(SpeakerPin);

 Serial.begin(250000); // 115200
 if ( digitalRead(GsmYnPin) == HIGH ) { // No GSM Shield when no Jumper [ Ohne GSM Shield, wenn der Jumper nicht gesetzt ist]
   GsmYn = 0;
 }

 if ( GsmYn == 1 ) {  // Only if GSM Shield is connected  [Nur wenn ein GSM shield vorhanden ist]
  // digitalWrite(ModemResetPin,LOW);
  Serial.println("SMS Messages Sender");
  boolean notConnected = true;  // connection status   [Verbindungsstatus]

  while (notConnected)
  {
    if (gsmAccess.begin(PINNUMBER) == GSM_READY) {
      notConnected = false;
```

```
    }
    else
    {
     Serial.println("Not connected");
     tone(SpeakerPin, 3000, 1000);
     delay(1000);
    }
   }
  Serial.println("GSM initialisiert");
  tone(SpeakerPin, 100, 1000);
  digitalWrite(RedLedPin, HIGH);
  //SmsText = "Das Epidino wurde eingeschaltet";  // "The Epidino has been switched on";
  //sendSMS();

  vvv = analogRead(Pot4Pin);
  www = 1;
  if ( vvv < 50 ) {                              //  tel#1
    tone(SpeakerPin, 500);
    delay(400);  // warten
    noTone(SpeakerPin);
    if ( tel == 2 ) { //
      www = 2;  // Es soll auch einen Ton fuer tel#2 geben
    }
    delay(400);  // warten
  }
  if (( vvv > 200 && vvv < 350 ) || www == 2 ) { //  tel#2
    tone(SpeakerPin, 600);
    delay(400);  // warten
    noTone(SpeakerPin);
  }
  if (( vvv > 500 && vvv < 660) || www == 3 ) {  //  tel#3
    tone(SpeakerPin, 700);
    delay(400);  // warten
    noTone(SpeakerPin);
  }
  if (( vvv > 800 ) || www == 4 ) {             //  tel#4
    tone(SpeakerPin, 800);
    delay(400);  // warten
    noTone(SpeakerPin);
  }

 }  // Ende wenn GsmYn = 1
 else
 { // When GSM shield is not used
  tone(SpeakerPin, 800);
  delay(100);  // warten
  noTone(SpeakerPin);
  tone(SpeakerPin, 500);
  delay(100);  // warten
  noTone(SpeakerPin);
  tone(SpeakerPin, 800);
  delay(100);  // warten
  noTone(SpeakerPin);
  tone(SpeakerPin, 500);
  delay(100);  // warten
  noTone(SpeakerPin);
 }
 Reset();
}

void loop() {
 StopButton();
 AlarmIN();
 AlarmSOS();
 AlarmShreshold = analogRead(Pot1Pin) / 30 ; // ..34s
 Duration = analogRead(Pot2Pin) / 200 ;      // ..5s
 Sens1Threshold = analogRead(Pot3Pin) ;

 if ( ST == 0 ) {   // Initial status. Not active.
  digitalWrite(RedLedPin, LOW);
  digitalWrite(GreenLedPin, LOW);
   LD = 0;
 }
 if ( ST < 9 ) {
  ReadSensor();  //  No alarm triggered yet [Alarm noch nicht ausgeloest]
 }
```

```
 if ( ST == 2 ) {       // Sensor activated [ Sensor ist aktiviert]
  digitalWrite(RedLedPin, HIGH);
  digitalWrite(GreenLedPin, LOW);
  ++LD;
 }
 if ( LD == 50 ) {
  digitalWrite(RedLedPin, LOW);
  digitalWrite(GreenLedPin, HIGH);

  LD = 0;
  ST = 1;
 }
 if ( ST == 9 ) { // Alarm triggered [Alarm ausgeloest] Alarm!
  if ( BZ < 5000) {
   digitalWrite(RedLedPin, HIGH);
   digitalWrite(GreenLedPin, LOW);
  }

  if ( GsmActive == 0 && RelayActive == 0 ) {      // Waiting loop for Stop Button [ Warteschleife fuer Stoptaste ]
    vvv = 0;
    while (vvv < AlarmDelay) { // Loop for Stop button [Warteschleife fuer Stopptaste]
     tone(SpeakerPin, 800); // Ton waehrend dem ein Fehlalarm gestoppt werden kann.
     StopButton();
     if ( ST == 0 ) {
      vvv = AlarmDelay;  // Stop button is pressed [Stopptaste gedrueckt]
     }
     ++vvv;
    }
    noTone(SpeakerPin);
  }

  if ( ST == 9 ) { // If Stop Button was not pressed

    if ( GsmYn == 1  &&  GsmActive == 0 ) {   // if GSM shield is used and no SMS was sent yet [GSM Shield, noch keine SMS geschickt]
     sendSMS();
    }
    if ( RelayActive == 0 ) { // if Alarm Relay is not activated yet [Alarm-Relais, noch nicht aktiviert]
     RelayOn();
    }
    ++BZ;
    if ( BZ == 5000) {
     digitalWrite(RedLedPin, LOW);
     digitalWrite(GreenLedPin, LOW);
    }
    if ( BZ == 10000) {
     tone(SpeakerPin, 100, 40); // short beep to signal that an SMS has been sent out
     //   digitalWrite(RedLedPin, HIGH);
     //   digitalWrite(GreenLedPin, LOW);
     BZ = 0;
    }
  }
 }
}

void ReadSensor() {
 Sens1Val = 0;
 vvv = analogRead(Sens1Pin);
 Sens1Val = abs(vvv);

 if ( mon1 == 1 ) {
  Monitor1();
 }

 if  ( Sens1Val > Sens1Threshold ) {
  if ( ST == 0 ) {   // First sensor activity
    TimerOn();
  }
  else     // Innerhalb der WartePhase  (not first sensor activity)
  {
    TimerEnd = millis() / 1000 + Duration;  // new TimerEnd
    ST = 2; // ST=2  must be before TimerCheck() [ ST=2 muss vor TimerCheck() sein.]
  }
 }
 if ( ST == 2 || ST == 1 ) {
  TimerCheck();
 }
```

```
}

void TimerOn() {     //  first sensor activation  [bei der ersten Aktivierung des Sensors ]
  TimerStart = millis() / 1000; // store actual time [aktuelle Zeit abspeichern]
  TimerEnd = TimerStart + Duration;  // First TimerEnd
  TimerAlarm = TimerStart + AlarmShreshold; // TimerAlarm

  ST = 2;

  digitalWrite(GreenLedPin, HIGH);
  digitalWrite(RedLedPin, LOW);
}

void TimerCheck() {
  if (  millis() / 1000 > TimerEnd ) { // If the Waiting Phase is over. [ Wenn die Wartephase abgelaufen ist, ]
    Reset();  //   ST=0
  }
  if  ( millis() / 1000 == TimerStart + AlarmShreshold -  2 * Duration) { // Warning [ Vorwarnung ]
    tone(SpeakerPin, 1000, 1);
  }
  if  ( millis() / 1000 > TimerAlarm )   { // If the timer ran longer then the AlarmShreshold. [ Wenn der Timer ueber die AlarmShreshold
gelaufen ist. ]
    Reset();
    ST = 9;                  // Alarm! Activity LED leuchtet stetig
    SmsText = "Name (Epidino)";
  }
}

void AlarmIN() {
  if ( digitalRead(AlarmINpin) == LOW ) { // Alarm by connected device like Epi-Care
    digitalWrite(GreenLedPin, HIGH);
    digitalWrite(RedLedPin, LOW);
    ST = 9;                  // Activity LED is lit steadily.  [Aktivitaets-LED leuchtet stetig]
    SmsText = "Name (Epi-Care)";
  }
}

void AlarmSOS() {
  if ( digitalRead(SosPin) == LOW ) { // Alarm by SOS button
    digitalWrite(GreenLedPin, HIGH);
    digitalWrite(RedLedPin, LOW);
    ST = 9;                  // Activity LED is lit steadily.  [Aktivitaets-LED leuchtet stetig]
    SmsText = "Name (SOS)";
  }
}

void StopButton() {
  if ( digitalRead(StopButtonPin) == LOW ) { //  StopButton was pushed. [ Die Stopptaste wurde gedrueckt. ]
    vvv = 0;
    if ( RelayActive == 1 || GsmActive == 1 ) {
      vvv = 1;
    }
    digitalWrite(ResetRelayPin, LOW); // Switch off Epi-Care unit
    Reset();
    delay(600); //200 zu wenig   500 ist ok
    digitalWrite(ResetRelayPin, HIGH);
    melodyStopp();
    delay(ResetPause); // wait to ovoid alarm triggered by the output signal of the Epi-Care unit
    //                  [Warten, um Alarmausloesung durch das Ausgangssignal vom Epi-Care zu verhindern.]
    tone(SpeakerPin, 100, 1000);
  }
}

void Reset() {
  digitalWrite(GreenLedPin, LOW);
  digitalWrite(RedLedPin, LOW);
  ST = 0;
  BZ = 0;

  digitalWrite(AlarmRelayPin, HIGH); // Switch off alarm relay immediately [Das AlarmRelais sofort abschalten]
  GsmActive = 0;   // GSM shield is not active [GSM Shield ist nicht aktiv]
  RelayActive = 0;  // Alarm relay is not active [AlarmRelais ist nicht aktiv]
  //  digitalWrite(ModemResetPin, LOW);
}

void sendSMS() {
```

```
 digitalWrite(GreenLedPin, LOW);
 digitalWrite(RedLedPin, HIGH);
 Serial.println("SENDING");
 Serial.println("SMS an gewaehlte Tel#: ");
 Serial.println("SENDING");
 Serial.println();
 Serial.print("Message:");
 Serial.println(SmsText);
 noTone(SpeakerPin);

 vvv = analogRead(Pot4Pin);  // 0 to 5 volts, digital value between 0 and 1023
 /*
 Ohne Zeitschaltuhr: Je nach Position von Pot4:
 Position linke untere Ecke (GND):   tel#1
 Position linke obere Ecke:          tel#2
 Position rechte obere Ecke:         tel#3
 Position untere rechte Ecke:        tel#4

 Mit Zeitschaltuhr: Pot4 auf Position linke obere Ecke auf tel#2
 Schaltuhr eingeschaltet = linke untere Ecke:   tel#1
 Schaltuhr ausgeschaltet = linke obere Ecke:    tel#2

 Wenn die Zeitschaltuhr das Relais eingeschaltet hat, wird tel#1 gewaehlt
 Wenn tel = 2 ist, dann wird auch tel#2 gewaehlt, wenn Pot4 auf linke untere Ecke gestellt ist oder die Schaltuhr das Relais eingeschaltet hat.
 */

 www = 1;
 if ( vvv < 50 ) {                            //  tel#1
   sms.beginSMS(tel1);
   SmsText = "tel#1"; // nur fuer testzwecke
   sms.print(SmsText);
   sms.endSMS();
   if ( tel == 2 ) { //
     www = 2;  // Es soll auch tel#2 angerufen werden
   }
 }
 if (( vvv > 200 && vvv < 350 ) || www == 2 ) { //  tel#2
   sms.beginSMS(tel2);
   SmsText = "tel#2"; // nur fuer testzwecke
   sms.print(SmsText);
   sms.endSMS();
 }
 if (( vvv > 500 && vvv < 660) || www == 3 ) { //  tel#3
   sms.beginSMS(tel3);
   SmsText = "tel#3"; // nur fuer testzwecke
   sms.print(SmsText);
   sms.endSMS();
 }
 if (( vvv > 800 ) || www == 4 ) {            //   tel#4
   sms.beginSMS(tel4);
   SmsText = "tel#4"; // nur fuer testzwecke
   sms.print(SmsText);
   sms.endSMS();
 }

 delay(1);
Serial.println("\nCOMPLETE!\n");
 GsmActive = 1;
}

void RelayOn() {  // Activation of relay [Aktivieren des Relay ] (ST = 9)
 digitalWrite(AlarmRelayPin, LOW); // Relay einschalten (Relay schaltet, wenn Minus)
 RelayActive = 1;  // Relay is activated [AlarmRelais ist aktiviert]
}

void RelayOff() {  // Deaktivieren des Relay nach einer vorgegebenen Zeit
 vvv = 0;
 while (vvv < RelayOnDuration) {
   StopButton();
   if ( ST == 0 ) {
     vvv = RelayOnDuration;   // Wenn die StopButton gedrueckt wurde, Schleife sofort verlassen
   }
   tone(SpeakerPin, 1200);
   ++vvv;
 } // End of loop [Schleifenende]
```

```
 digitalWrite(AlarmRelayPin, HIGH); // turn off relay  [Relay abschalten (Relay schaltet, wenn Minus)]
 RelayActive = 5;  // AlarmRelais ist nach der vorgegebenen Zeit deaktiviert worden
 noTone(SpeakerPin);
}

void melodyStopp() {
  if ( vvv == 0 ) { // RelayActive = 0
    tone(SpeakerPin, 300);
    delay(800);  // warten
    tone(SpeakerPin, 1000);
    delay(800);  // warten
    noTone(SpeakerPin);
  }
  if ( vvv == 1 ) { // RelayActive = 1
    tone(SpeakerPin, 1000);
    delay(1000);  // warten
    noTone(SpeakerPin);
    vvv = 0;
  }
}

void Monitor1() {

 Serial.print("Vers.");
 Serial.print(verNo);

 Serial.print("    TimerStart:");
 Serial.print(TimerStart);

 Serial.print(" Time:");
 vvv = millis() / 1000;
 Serial.print(vvv);

 Serial.print(" TimerEnd:");
 Serial.print(TimerEnd);

 Serial.print(" AlarmShreshold:");
 Serial.print(AlarmShreshold);

 Serial.print(" Duration:");
 Serial.print(Duration);

 Serial.print(" Sens1Threshold:");
 Serial.print(Sens1Threshold);

 Serial.print(" Sens1Val:");
 Serial.println(Sens1Val);
}
```