Sa’Bi Baby Care System

Yerlan Amanzholov[1], Timur Malikov[1], Aisultan Alimkhan[1], Aigerim Iskakova[1], Askar Maxim[1]

***Abstract*— The project introduces the Sa’Bi baby care system. The system is built using Raspberry Pi single-board computer and Arduino microcontroller. Using this two as its core elements, the device that can gather a data about an environment that the child is currently surrounded by, and send the data via internet to a Telegram bot. Practical experiment showed that the sensors work well and device responds to the commands received from the user instantly. The paper describes the working principles of two core parts and makes suggestions on future improvements.**

I. Introduction

The advent of modern technologies changed our everyday life and made it way more convenient than it was in the past. However, even with all these technological miracles the child caring hardly ever changed. Bringing up a future generation in a safe environment was always prioritized from both moral and instinctual point of view, especially during the early development periods, when they cannot take care of themselves. This means they require constant attention of their parents and this in turn brings its own demerits. For instance, increased stress, lack of sleep, and as a result of it drop in productivity, which affects financial situation of a family. Going through this period takes its toll on parents, and while in some cultures, were it is common for grandparents (i.e. Kazakh culture) to live with newlyweds under the same roof, they can help to easy the burden, for others there are numerous available options on the market such as, walkie-talkies that can notify parent if the baby is crying, or video monitoring systems, or hiring a nanny. Although all of them seem plausible, they all possess either financial or practical constraints that make them ”not bad, but not good enough” options. This paper is going to introduce a new innovative design of a system called ”Sa’Bi” that is based on Arduino micro-controllers and can be easily accessed from any point inside or outside of the house with your smartphone.

II. Arduino

In this section circuit configuration will be discussed. Each sensor connection will be described separately.

Devices and sensors

1) Microcontroller Arduino UNO R3
2) Breadboard 830 pins
3) Sound detector
4) Photoresistor
5) Temperature and humidity sensor DHT11
6) Resistors
7) Wires
8) Arduino connector cable

Sound detector Sound detector used to detect the noises in baby room.

Pins of the sensor	Pins of Arduino
Vcc	5V
GND	Ground
OUT	Digital Pin 2

TABLE I: Pin configuration of the sound sensor

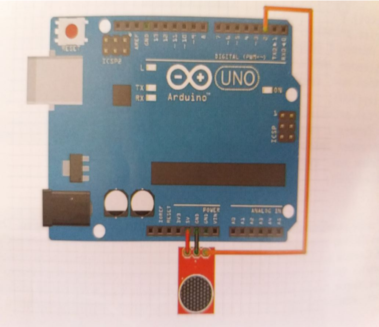


Fig. 1: Sound detector sensor connection

```
int soundPin = 13;
soundOut = digitalRead(soundPin);
if(soundOut == HIGH){
Serial.println("cry");
delay(1000);
```

Code given above represents the operation principle of the sound detector. If soundOut goes high, then ”cry” signal is sent through serial port every 1 sec.

Photoresistor

Photoresistor is used for verifying the light level in the baby room. In this project 3 photoresistors were used for checking the light from 3 sides simultaneously.

Inputs ldr1 (A0 pin), ldr2 (A1 pin), ldr3 (A3 pin) represents the values of 3 photoresistors. The Code below shows that Arduino reads the analog values of the photoresistors, converts the voltage value into Lux, which is a measure of light intensity. Then Arduino sends the values into the serial port. The code for one photoresistor is provided below

```
intensity1 = analogRead(ldr1);
float volts1 = intensity1 * 0.0048828125;
int lux1=500/(10*((5-volts1)/volts1));
Serial.println(int(lux1));
```

*This work was not supported by any organization

[1] A. Maxim, Y. Amanzholov, T. Malikov, A. Alimkhan, A. Iskakova are with Faculty of Electrical and Electronics Engineering, Nazarbayev University, 010000 Astana, The Kazakhstan

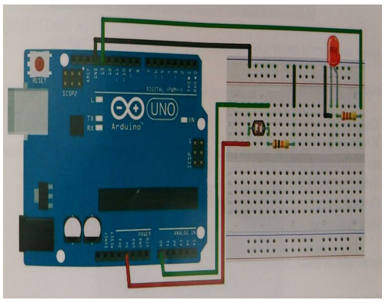


Fig. 2: Photoresistor connection

Temperature and humidity sensor DHT11 Sensor DHT11 detects the temperature and humidity of the baby room.

Pins	Arduino pins
Vcc	5V
OUT	Pin 7
GND	grounding

TABLE II: Pin connection of the DHT11 sensor

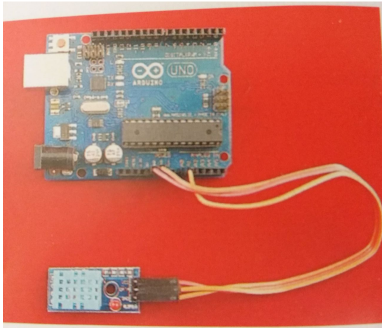


Fig. 3: Sensor DHT11 connection

After the humidity and temperature reading recordings system prints their values. However, if there is an error in reading of the temperature and humidity the system notifies it.

```
if (isnan(event.temperature)) {
   Serial.println("Error reading
       temperature!");
}
else {
   Serial.println(event.temperature);
}

if (isnan(event.relative_humidity)) {
   Serial.println("Error reading humidity!");
}
else {
   Serial.println(event.relative_humidity);
}
```

III. Raspberry Pi

After successfully assembling Sa'Bi's hardware and writing main functions in Arduino's special software, a single-board computer Raspberry Pi was used in order to allow users to receive notifications wirelessly. System itself requires power, LAN connection and hardware. An access to Raspberry can be obtained with usage of SSH cryptographical network protocol, to be more specific, ”-ssh pi@ipaddress” code line. This was followed by the creation of the telegram bot for the baby caring system via @BotFather on telegram. In order for Sa'Bi bot to function, python platform was used, while main code is available in the appendix. Considering software, there are 5 functions which were created and are accessible for potential users.

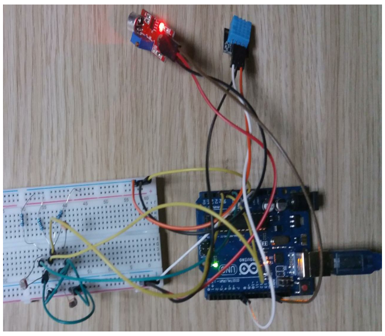
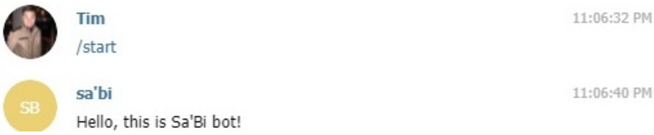


Fig. 4: Circuit connection of Arduino part of the project

A. ”/start” Command

”/start” command allows user to switch on the system. As soon as the bot is working, a message ”Hello, this is Sa'Bi bot!” welcomes the user and notifies, that system is ready to operate.



Tim 11:06:32 PM
/start

sa'bi 11:06:40 PM
Hello, this is Sa'Bi bot!

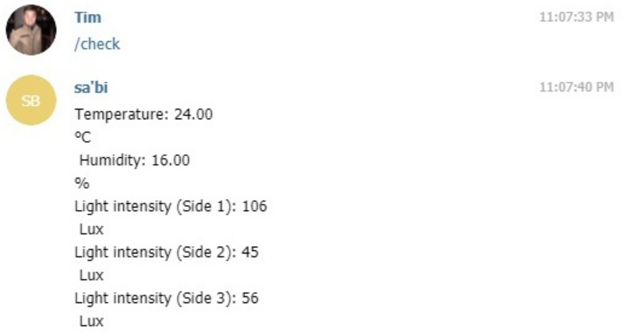
Fig. 5: /start command

B. ”/check” Command

”/check” command is consisted of various sublines which have direct relation to the sensors of the Sa'Bi. Variable ”t” determines the temperature read via sensor, 'h' stands for humidity and light1, light2 and light3 are the light intensity of sides 1,2 and 3 respectively. ”/check” calls the outlined parameters and displays them in one message.

C. ”/recommend” Command

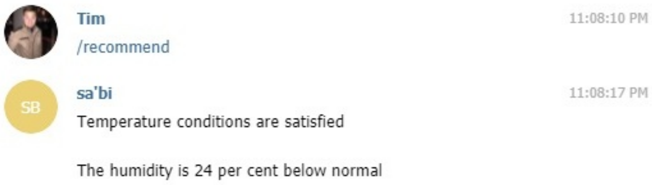
”/recommend” command compares the current values of temperature and humidity with those, that were set as ”normal” by the developers. In this case, standard temperature lays between 21 and 24 decreases, whereas normal values for the latter are in range of 40 and 60. In case numbers are less, or greater than standard, system will display by how much percent the conditions are below or above normal.



Tim 11:07:33 PM
/check

sa'bi 11:07:40 PM
Temperature: 24.00
°C
 Humidity: 16.00
%
Light intensity (Side 1): 106
 Lux
Light intensity (Side 2): 45
 Lux
Light intensity (Side 3): 56
 Lux

Fig. 6: /check command



Tim 11:08:10 PM
/recommend

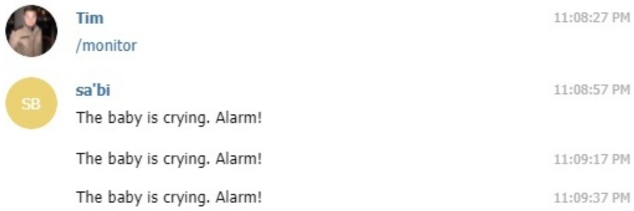
sa'bi 11:08:17 PM
Temperature conditions are satisfied

The humidity is 24 per cent below normal

Fig. 7: /recommend command

D. "/monitor" Command

"/monitor" command is used for determining whether the baby is crying or not. To be more specific, it monitoring the noise in the self-titled sensor. As soon as it detects the sound, raspberry pi notifies the user that their baby is crying via "The baby is crying. Alarm!" message.



Tim 11:08:27 PM
/monitor

sa'bi 11:08:57 PM
The baby is crying. Alarm!

The baby is crying. Alarm! 11:09:17 PM

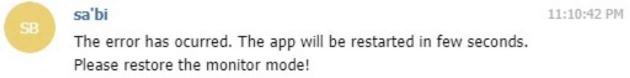
The baby is crying. Alarm! 11:09:37 PM

Fig. 8: /monitor command

E. "/stop" Command

"/stop" command shuts down the monitor mode.

Notwithstanding the fact, that system is working perfectly, there is still a possibility, that Sa'Bi will crush. In emergency situations, it recalls for the restart program function, which forces the system to start from scratch.



sa'bi 11:10:42 PM
The error has ocurred. The app will be restarted in few seconds.
Please restore the monitor mode!

Fig. 9: Warning notification

IV. Recommendations

The Sa'Bi baby care system allows parent to monitor the conditions of a child from the distance. However, following recommendations for further improvement of the system can be made.

- Increase the number of bot commands. For instance, bot command to check average temperature over one day can be added.
- Add the Bluetooth module to the system. By Bluetooth connection, the system can be controlled in the absence of internet connection. However, the range of such connection is limited.
- Voice control can be added. This allows the user not to waste time to type the commands.
- Power supply. In this project, Arduino and Raspberry Pi were powered from laptop. Separate power supply can be purchased to make the system autonomous.
- During the project, Raspberry Pi 2 was used for connection with Telegram bot. This generation of Raspberry Pi computers can only be connected to the internet by ethernet cable. This seriously limits the areas, in which the device can be placed. To resolve this problem, network interface card in form of flash drive can be purchased to connect the device to the internet through the Wi-Fi. The other solution can be to buy new generation of Raspberry Pi, which supports the Wi-Fi connection.
- Connect the system to other devices in the room. For examples, air fan can be automatically switched on and off to keep the temperature in the room on the recommended level.
- The system can be made waterproof, since there is a big chance that water will be spilled on the device.
- Since it is very costly to develop Telegram bot for every new user, unique ID should be assigned to every user. By this ID, user can work with bot. So bot code should undergo major improvements to start global manufacturing of this product. Furthermore, security measures need to be considered, since leak of users ID can disclose confidential information.

V. Conclusion

As it was outlined above, provided paper propounds an innovative, easy to manufacture and cost-effective solution for distant monitoring of environmental conditions around your infants. Device itself allows parents to track the temperature, humidity and light intensity while doing their own business, whereas specially created "Sa'bi" bot notifies them whenever their baby is making any noise. The whole project was executed by combination of Raspberry Pi and Arduino microcontroller. Owning to user friendly interface, users will get used to Sa'bi immediately, that will fully open up the potential of the baby caring system. In addition, with further improvements, system can be easily implemented into smart houses, which will allow users to remotely maintain the ideal environmental conditions not only around their children, but whole house as well. In the upcoming future, provided system can easily become a vital part of any house.

APPENDIX

A. Arduino Code

```
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#define DHTPIN      7
#define DHTTYPE     DHT11

DHT_Unified dht(DHTPIN, DHTTYPE);

int ldr1 = A0;
int ldr2 = A1;
int ldr3 = A2;
int intensity1;
int intensity2;
int intensity3;
int soundPin = 13;
int soundOut;

void setup() {
  Serial.begin(9600);
  dht.begin();
  pinMode(ldr1, INPUT);
  pinMode(ldr2, INPUT);
  pinMode(ldr3, INPUT);
  pinMode(soundPin, INPUT);
}

void loop() {
  String input = "";
  while (Serial.available() > 0) {
  input += (char) Serial.read();
  delay(5);
  }
  if(input=="t"){
    sensors_event_t event;
    dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println("Error reading
        temperature!");
    }
  else {
  Serial.println(event.temperature);
    }
  }
  else if(input=="h"){
    sensors_event_t event;
    dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity)) {
    Serial.println("Error reading
        humidity!");
    }
  else {
    Serial.println(event.relative_humidity);
    }
  }

  else if(input=="l1"){
    intensity1 = analogRead(ldr1);
    float volts1 = intensity1 * 0.0048828125;
    int lux1=500/(10*((5-volts1)/volts1));
    Serial.println(int(lux1));
  }

  else if(input=="l2"){
    intensity2 = analogRead(ldr2);
    float volts2 = intensity2 * 0.0048828125;
    int lux2=500/(10*((5-volts2)/volts2));
    Serial.println(int(lux2));
  }

  else if(input=="l3"){
    intensity3 = analogRead(ldr3);
    float volts3 = intensity3 * 0.0048828125;
    int lux3=500/(10*((5-volts3)/volts3));
    Serial.println(int(lux3));
  }

soundOut = digitalRead(soundPin);
if(soundOut == HIGH){
  Serial.println("cry");
  delay(1000);
  }
}
```

B. Raspberry Pi

```
import telepot
import serial
import time
import apscheduler
from apscheduler.schedulers.background
import BackgroundScheduler
import logging
logging.basicConfig()
import os
import sys

ser=serial.Serial(’/dev/ttyACM0’,9600)

def restart_program():
  python = sys.executable
  os.execl(python, python, *sys.argv)

def condition(chat_id):
  check = ser.readline()
  if "cry" in check:
    bot.sendMessage(chat_id,"The baby is
        crying. Alarm!")
    ser.flushInput()
    ser.flushOutput()
  else:
    bot.sendMessage(chat_id, " The error has
        ocurred. The app will be restarted
        in few seconds. Please restore the
        monitor mode!")
    ser.flushInput()
    ser.flushOutput()
    restart_program()


sched=BackgroundScheduler()
sched.start()

def handle(msg):
  content_type, chat_type, chat_id =
      telepot.glance(msg)
  chat_id = msg[’chat’][’id’]
  command = msg[’text’]
  if command == ’/start’:
    time.sleep(1)
    bot.sendMessage(chat_id, 'Hello, this is
        Sa\'Bi bot!')
  elif command == '/check':
    message=""
    ser.flushInput()
    ser.flushOutput()
    ser.write('t')
    temp = "Temperature: " + ser.readline()
        + u"\u2103 \n "
    ser.flushInput()
    ser.flushOutput()
    message=message+temp
    ser.flushInput()
    ser.flushOutput()
    ser.write('h')
    hum = "Humidity: " + ser.readline() + "%
        \n"
    ser.flushInput()
    ser.flushOutput()
    message = message + hum
    ser.flushInput()
    ser.flushOutput()
    ser.write('l1')
    light1 = "Light intensity (Side 1): " +
        ser.readline() + " Lux \n"
    ser.flushInput()
    ser.flushOutput()
    message = message + light1
    ser.flushInput()
    ser.flushOutput()
    ser.write('l2')
    light2 = "Light intensity (Side 2): " +
        ser.readline() + " Lux \n"
    ser.flushInput()
    ser.flushOutput()
    message = message + light2
    ser.flushInput()
    ser.flushOutput()
    ser.write('l3')
    light3 = "Light intensity (Side 3): " +
        ser.readline() + " Lux \n"
    ser.flushInput()
    ser.flushOutput()
    message = message + light3
    bot.sendMessage(chat_id, message)

  elif command == '/monitor':
    sched.add_job(condition, 'interval',
        [chat_id], seconds=20,
        max_instances=1000, id = '123')
  elif command == '/stop':
    try:
      sched.remove_job('123')
      ser.flushInput()
      ser.flushOutput()
    except:
      bot.sendMessage(chat_id, "The monitor
mode was not started!")
  elif command == '/recommend':
    message = ""
    ser.flushInput()
    ser.flushOutput()
    ser.write('t')
    temp = ser.readline()
    ser.flushInput()
    ser.flushOutput()
    if (float(temp)<21):
      tdifference = 21 - float(temp)
      message = message + "The temperature
          is %d C below
          normal\n\n"%tdifference
    elif (float(temp)>24):
      tdifference = float(temp) - 24
      message = message + "The temperature
          is %d C above
          normal\n\n"%tdifference
    else:
      message = message + "Temperature
          conditions are satisfied\n\n"
    ser.flushInput()
    ser.flushOutput()
    ser.write('h')
    hum = ser.readline()
    ser.flushInput()
    ser.flushOutput()
    if (float(hum) < 40):
      hdifference = 40 - float(hum)
      message = message + "The humidity is
          %d per cent below
          normal\n"%hdifference
    elif (float(hum) > 60):
      hdifference = float(hum) - 60
      message = message + "The humidity is
          %d per cent above
          normal\n"%hdifference
    else:
      message = message + "Humidity
          conditions are satisfied\n"
      bot.sendMessage(chat_id, message)

  else:
    bot.sendMessage(chat_id,'Wrong command')

bot = telepot.Bot('539401906:
AAE5a7LY5ghahSx-jMRjDVVgpgQ4Sr58TMM')
chat_id = bot.message_loop(handle)
print 'Bot for sa\'bi baby care system.'

while 1:
time.sleep(10)

#Username:the_sabi_bot
#Token:539401906:AAE5a7LY5ghahSx-jMRjDVVgpgQ4Sr58TMM
```