White Hand

Contents

Project within the framework of the course "Applications of Telecommunications Devices"

Authors: Kampouridou Iliana, Papakostas Konstantinos

Supervising professor: Dimitriou Antonis

May 2023, Thessaloniki

ARISTOTLE UNIVERSITY OF THESSALONIKI

Import

As part of the course on telecommunications device applications, a sign language translation glove was implemented. This is an application that recognizes the speaker's hand movement and displays the content of his speech to the recipient, who decides.

Network design analysis

The application study was carried out in the context of a conference of deaf-mute people consisting of a total of 2000 guests. A thousand of them are speakers and a thousand of them are deaf-mute. The central node (base-station) will receive messages from 1000 transmitters and will route them, at most, to 1000 receivers. It is recalled that the ALOHA protocol has been used for the networking of the nodes. Therefore, with the above data, it follows that the total telecommunications traffic in the network is rτ = 1000*3/s*8bit/125Mbit = 0.000192 and finally the total number of channels required for the proper operation of the network is channels = rτ / 0.186 = 0.0010, i.e. one channel.

Implementation

For the implementation we used 3 different types of nodes:

- The transmission node is connected to the glove and sensors. It collects and interprets the data and then sends it to the central node. Each deaf person using a glove corresponds to such a node. It uses 2 flex sensors to determine the positions of the thumb and index finger and a button to set the receiver address.

- The receiving node is connected to the displaylcd. It receives the data from the central node and prints it on the screen. Each person who does not know sign language corresponds to such a node.

- The central node processes the data and communicates with everyone the nodes. It ensures that the data of each transmitter reaches the correct receiver. It works alternately as a receiver and a transmitter.

Connections

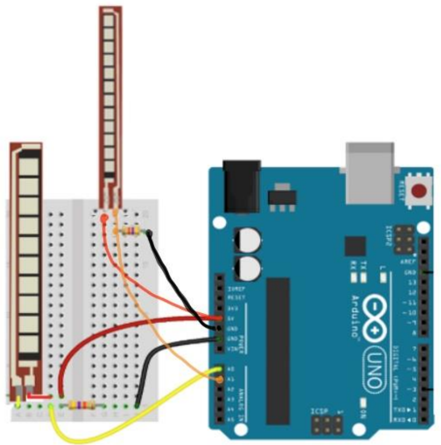

Figure 1: Transmission Node Connections

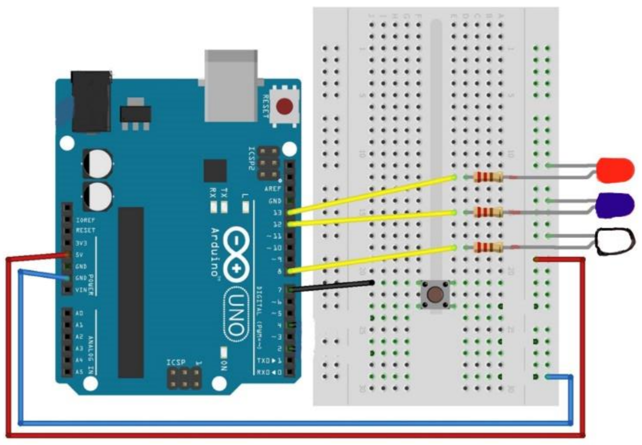

Figure 1: Transmission Node Connections

Figure 2: Central Node Connection

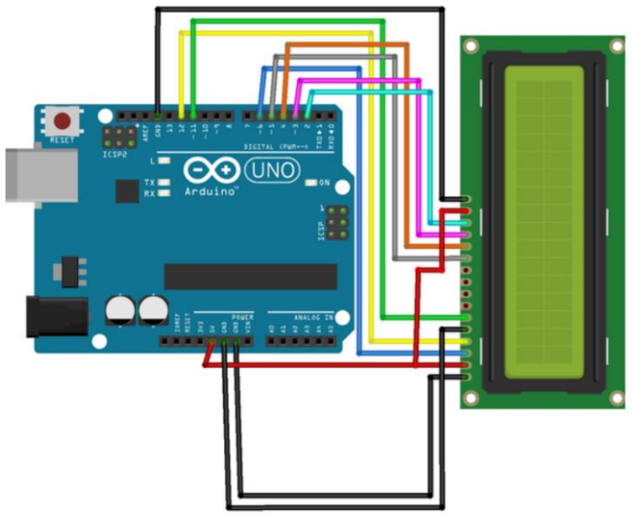

Figure 3: Receiving Node Connection

Transmission node

Starting with the pins for connecting the sensors, all global variables related to communication and the button are initialized.

In the setup function, the pins of the 2 flex sensors and the button are defined as inputs. Communication is established with a selected baud rate of 9600 bps, communication frequency of 425 MHz, power of 20 dbm and pulse modulation of FSK_Rb125Fd125 and the central loop address is added to the route.

In the loop function, the packet to be sent to the main loop is created. First, we check if the button was pressed by reading the buttonPin. If it was pressed, we toggle the value of the address to send the packet send_to (in the example between 7 and 8).

Then we read the values   of the flex sensors for the thumb and index finger (respectively it can be extended to the other fingers). Depending on the values   we encode into a specific letter, as they result from the position of the hand for the representation in ASL (note the calibration is repeated for a different person or movement of the relative position of the sensor). Otherwise the result is interpreted as an idle state or we detect a connectivity error. The letters are encoded as follows

- - 3: error code for disconnection of both sensors
- - 2: error code for thumb sensor disconnection
- - 1: indicator sensor disconnection error code
- 0: idle state.
- 1: letter A
- 2: letter B
- 3: letter C
- 4: letter D

as values   close to 0 imply disconnection of the cables. Each of these states implies a number which is stored in the variable data.

The variable info is the packet to be sent and combines the data information, with the id of the sender that sends the information as well as the id of the receiver in a three-digit number. The first digit refers to the information/letter to be transmitted, the second is the address of the sender and finally the last digit defines the receiving node that concerns the specific packet. For example, for sending the letter A from the sender with address 5 to the receiver with address 7, the packet to be sent that is formed has a value of 157. This value is sent to the central node. If the sending is unsuccessful, the sending is attempted after a delay of 100ms, while if it is successful, the repetition loop is completed after a delay of 50ms.

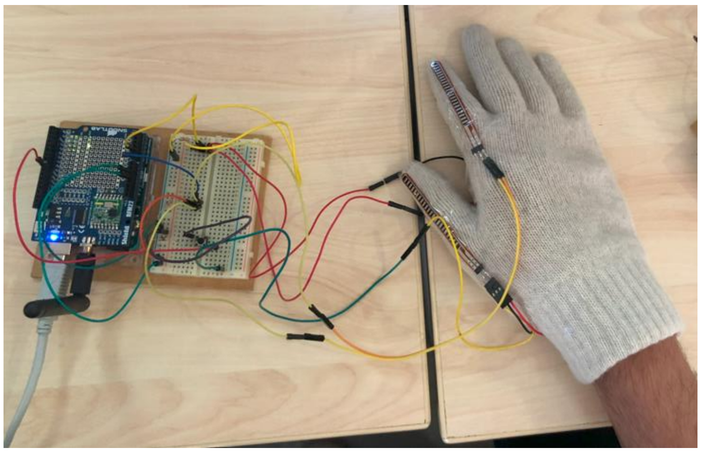

Figure 4: Transmitter implementation

Calibration

Before using the glove, it is required to perform a correct calibration on the hand of the person who is going to use it. After connecting the sensor pins to the node, we read them and print them on the screen. The hand that will wear the glove forms the letters one by one and based on the results we create the appropriate detection areas. It is noted that for the recognition of the letter, the values   of each sensor must belong simultaneously to both intervals for its correct translation.

Receiving node

In this node, the letter is decoded and printed on the screen. Initially, the lcd screen, the constants for communication and some constants for printing the information (data, info, node) are defined.

In the setup function, the screen is initialized, communication with similar characteristics to the receiver is established, and the address of the central node only is added to the route.

In the loop, the receiving process is prepared. The received value is placed in the variable received_value and from there we store it in info. The data consists of the letter that was formed and the address of the node that sent it. By using the operations of division and remainder, the information is isolated and each value is placed in the variable data and node respectively. For example, if the received value is 15, then dividing by 10 we find the letter that was sent, while the remainder of the division by 10 will give us the address of the sender.

When printing the information, the address from which the information was received appears on the top line of the screen and the letter or message that was sent to me appears on the bottom line (which is decoded from the data value with a multiple condition).

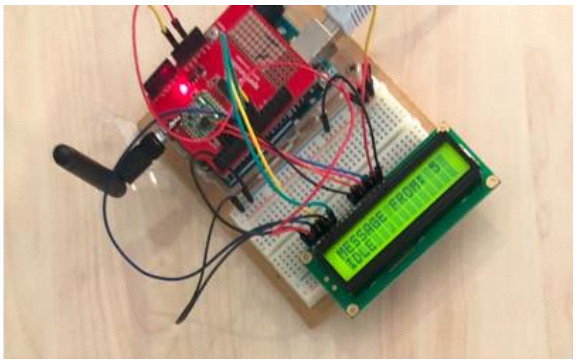

Figure 5: Receiver implementation

Central hub

At the central node, packets are received from all the talking nodes and routed to the appropriate recipients. Starting with, the desired addresses, variables for the operation of the status LEDs as well as variables for the management and storage of data are defined.

In the setup function, the communication is defined similarly to the other loops. Both the speaking convoys and all the recipients participating in the communication are added to the route. Finally, the LEDs are initialized by setting their pins as output and turning them off.

In the loop, packets are received and immediately forwarded to the appropriate receivers. This node alternates between being a transmitter and a receiver, this switching is shown in the node_mode variable. Initially, it functions as a receiver, preparing the reception process and waiting to receive data. Once it receives a packet, it transfers the value of the received_value variable to info. With appropriate use of the remainder of division, it separates the address of the receiver from the rest of the data. For better understanding, let's say that the packet received is 157. Dividing this number by 10 creates the packet and with the remainder of division by 10 gives us the receiver to which it will be forwarded.

Of these, only the information from the sensors is checked (without the addresses) and the corresponding LED lights up to indicate that it recognizes what it received (blue for correct letter reception, white for idle status and red for any error).

Upon receiving the packet, it momentarily switches to transmitter mode and transmits to the receiver that the transmitter has indicated the information from the variable data . If the communication fails, it tries again up to 20 times and then breaks. This is done to avoid the creation of an endless loop and the creation of huge up to infinite urine during communication. When the sending is achieved or a break occurs, the repetition structure is completed.

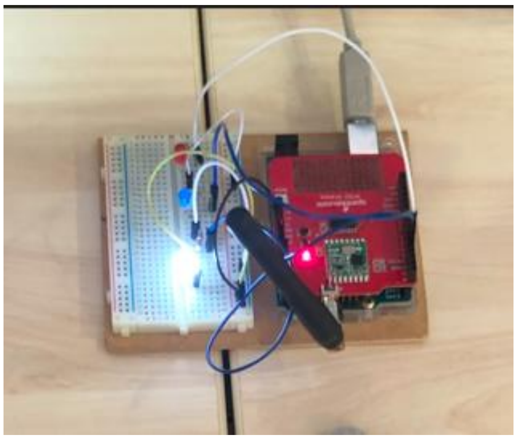

Figure 6: Implementation of the central node

Pseudocodes

Transmitter

```
Setupthe pins for the flex pins

Initialthe data value

Setupvariables for communication (my_address, sent to, info, successful_packet)

Setupvariables for button (buttonPin, button_state, buttonState)

SETUP:

Setthe pins for flex sensors and buttons as input
```

```
Setup the communication with speed 9600 bps, frequency 425 MHz, power 20 dBm and
modulation FSK_Rb125Fd125
Add to the route the central node


LOOP:
Set buttonState as value from the pin button
If buttonState is HIGH
        Change the send_to value to the next receiver address
Set ADCflexIn as the value of flexPin52 and ADCflexTh as the value of flexPin25
Print those variables
Encode sensors values   to a letter or a state
Print it on the screen and put it on variable data
If data<0
        info = data*100- my_Address*10 - send_to
else
        info = data*100 + my_Address*10 + send_to
Print the information
Create packet
Send info variable
While successful_packet is zero
        If packet fails to be sent
                Print it
                try again after 100 ms
        else
                Make successful_packet 1
                Print the success
Delay 50ms
```

Transceiver

```
Set up variables for the leds (ledPinW, ledPinB, ledPinR, ledOn, ledOff)

Initialize variables (data, successful_packet, node_mode, info, send_to, fail_counter,
data_leds )

SETUP:

Setup the communication with speed 9600 bps, frequency 425 MHz, power 20 dBm and
modulation FSK_Rb125Fd125

Add to route all the nodes of our network

Set up the led variables as output and off

LOOP:

If node_mode=0

        Print "receive mode"

        Make node_mode=1

Setup receiving procedure

Put received data in received_value variable

Print received_value

info=received_value

data = info/10

send_to = abs(info%10)

Print data and sent_to

data_leds = info/100

if data_leds>0

        turn only ledPinB on

else if data_leds=0

        turn only ledPinW on

else if data_leds<0 and data_leds>-4

        turn only ledPinR on

if info = received_value

        node_mode=0

        Make successful_packet false

        Print "transmitted mode"

        Setup transmission procedure
        Send data variable
        While successful_packet is zero
            If packet fails to be sent
                Print it
                Rise fail_counter by 1
                If fail_counter=2
                    fail_counter=0
                    break
                    try again after 30 ms
            else    Make successful_packet true
                    Print the success
        Delay 100ms
```

Receiver

```
Setup variable for lcd contrast
Setup LCD pins (9, 8, 5, 4, 3, 7)
Setup variables data, info, node
SETUP:
Setup LCD
Setup the communication with speed 9600 bps, frequency 425 MHz, power 20 dBm and
modulation FSK_Rb125Fd125
Add to route the central node
LOOP:
Setup receiving procedure
Put received data in received_value variable
info=received_value
data=info/10
node=abs(info%10)
Print node variable on the front line of lcd
Decode data
Print the appropriate message on the second line of lcd
Delay 10ms
```

```
Calibration
Setup the pins of the flex sensors

SETUP:
Start serial communication 9600

Set pin mode of the flex sensors

LOOP:
ADCflexIn = value of flexPin52

ADCflexTh = value of flexPin25

Print the values

Delay 1s
```

Related Links

Commercial:
https://youtu.be/EPcsYgDs9gM

Video Demo:
https://youtu.be/o5tppeNglUs

Function codes:
https://drive.google.com/drive/folders/18rS4uPgnhjyPJwzPFIw0I1OZvrJSezHE?usp=share_link

Bibliography

https://techatronic.com/final-year-project-for-ece-arduino-sign-language-glove/

https://lastminuteengineers.com/flex-sensor-arduino-tutorial/

https://www.instructables.com/Arduino-Interfacing-With-LCD-Without-Potentiometer/

Student Information

Full name : Kambouridou Iliana

A.E.M.: 10169

E-mail : iliakamp@ece.auth.gr

Full name : Papakostas Konstantinos

A.E.M. : 10127

E-mail : kdpapako@ece.auth.gr