Tester Ethernet Cable

Author Tunaru Vlad

Introduction

The project aims to create an Ethernet cable tester that displays the type of cable and its functionality. To create this tester, we were inspired by the testers that are currently on the market.

General Description

- The cable is composed of 8 wires, on which the signal is transmitted successively.
- On the other end of the cable, the signal is received and the order in which it arrived is recorded.
- The order is analyzed and the type of cable is determined.
- If no signal is received at the receiving end, an error message is displayed.

Ethernet cables are classified into 4 types based on the order of signal transmission and reception:

Ordered List ItemSTRAIGHT TROUGH cable:

```
Sender: [1 2 3 4 5 6 7 8]
Receiver: [1 2 3 4 5 6 7 8]
```

Ordered List ItemCROSSOVER T568 A cable:

```
Sender: [1 2 3 4 5 6 7 8]
Receiver: [3 6 1 7 8 2 4 5]
```

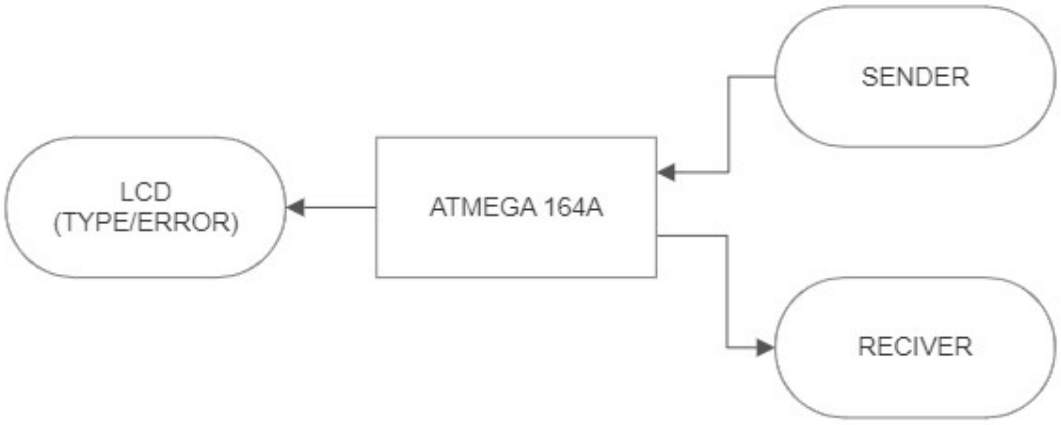
Unordered List ItemCROSSOVER T568 B cable:

```
Sender: [1 2 3 4 5 6 7 8]
Receiver: [3 6 1 4 5 2 7 8]
```

Ordered List ItemROLLOVER cable:

```
Sender: [1 2 3 4 5 6 7 8]
Receiver: [8 7 6 5 4 3 2 1]
```

The type and configuration of the cable will be displayed on the LCD screen. In case the cable is defective, it will be displayed using '0' instead of the faulty cable.



Hardware Design

- LCD screen
- 8 resistors 82k
- connector cables mother-father

For sender we used the PA pins of the microcontroller:

```
1-PA0, 2-PA1, 3-PA2, 4-PA3, 5-PA4, 6-PA4, 7-PA6, 8-PA7
```

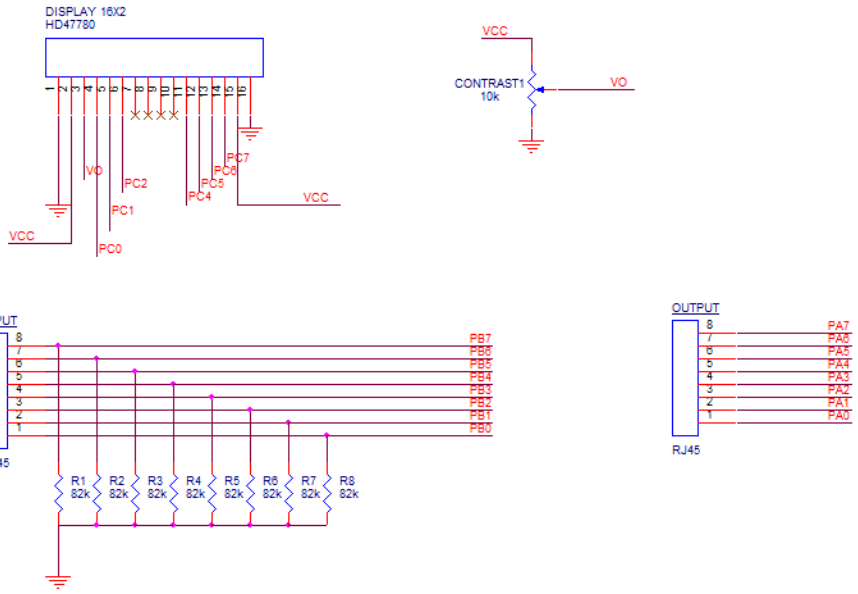
For receiver we used the PC pins of the microcontroller:

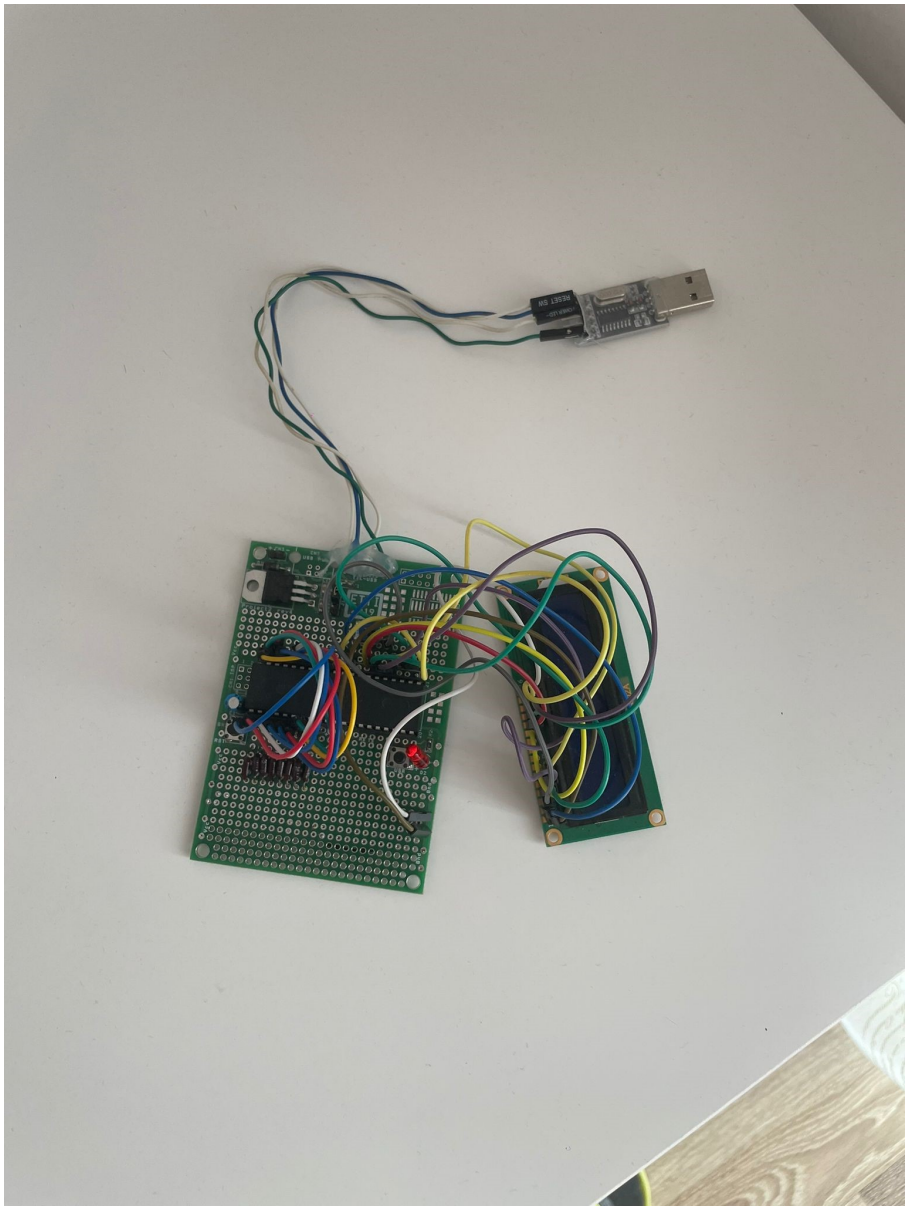
```
1-PB0, 2-PB1, 3-PB2, 4-PB3, 5-PB4, 6-PB5, 7-PB6, 8-PB7
```

To display the results on the LCD screen, we used the PC pins:

```
PC0, PC1, PC2, PC4, PC5, PC6, PC7
```

Hardware connections schematic in ORCAD:





Software Design

For this project, the following resources of the microcontroller were used:

• 1 general-purpose pin set as an output for an LED (PD6)

• 1 general-purpose pin set as an input for a button (PD5)

• Alphanumeric LCD display with HD44780 driver configured on port C

• 16 pins used as IN/OUT for cable continuity testing

The testing program uses 1 port with 8 bits for signal transmission and another port for reception, utilizing PORTA and PORTB, one defined as an input and the other as an output. Two arrays are used to store the display sequences for individual LED display or continuous display. For continuity testing, the individual display sequence is used, and the received result at the reception port is stored in the reception[8] array, thus preserving the order in which the test signal was received. By comparing the reception[8] array with three other arrays storing the correct sequences, the type of cable tested (crossover, rollover, straight) can be determined. If none of these patterns match, the cable is declared defective. The display of the detected cable type is done through an LCD display along with a map indicating the defective or good wires.

```
    for(i=0;i<=7;i++){ // iterate through each bit of the test port and turn
it on
    PORTA = afisare_single[i];
    }
```

This code snippet activates the pins on the test port. The logical signal 1 will pass through the cable and reach the reception port. Depending on the order in which it arrives, the values will be stored in an array to identify the type of cable.

```
  if(PINB > 0) {  // if something is detected at input on port B
  if(IN1 == 1) receptie[i] = 1;   // write the corresponding value for the
detected pin
  else if(IN2 == 1) receptie[i] = 2;
  else if(IN3 == 1) receptie[i] = 3;
  else if(IN4 == 1) receptie[i] = 4;
  else if(IN5 == 1) receptie[i] = 5;
  else if(IN6 == 1) receptie[i] = 6;
  else if(IN7 == 1) receptie[i] = 7;
  else if(IN8 == 1) receptie[i] = 8;
  }  // end if detection
```

The received values are stored in the receptie[] array, preserving the order of each measured pin number. This array can be directly used to display the values on the LCD screen.

```
  if(receptie[i] == straight[i]) sum++;
  if(sum == 8) {   // if all 8 wires are connected
  lcd_puts("Defect   ");
  for(i=0;i<=7;i++){     // print the reception vector
      lcd_gotoxy(i,1);
      itoa(receptie[i],buffer);
      lcd_puts(buffer);
  }
```

```
}
```

The checking for each type of cable is done by testing each value in the received vector. If all 8 values are correct, it means that all wires are good and the respective configuration has been detected. If not all values are correct, another configuration is tested, and if no cable is identified, it will be declared defective. For a defective cable, the detected order is still displayed to identify the interrupted wires.

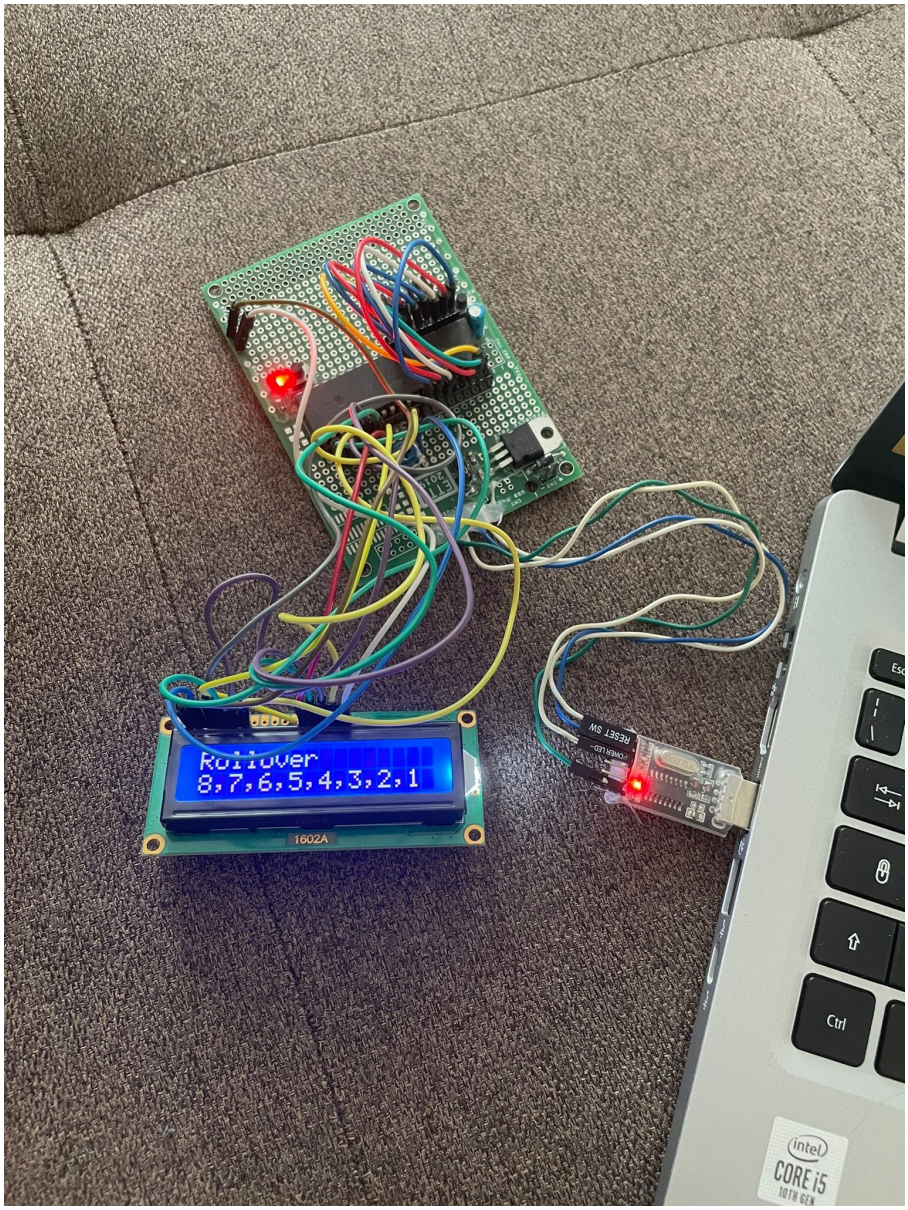
The system performs the testing only once upon request, by pressing the SW button. After a test, a new test can be initiated by pressing the button again, which resets all variables and starts a new measurement.

The display is done using a 2×16 character LCD screen with a built-in HD44780 controller. The display is powered from the system's 5V voltage, and the contrast level is set by a potentiometer. The display control is done using 7 bits connected to port C.

The user interacts with the system through a momentary push button. This button is read by the microcontroller on an input pin, which is maintained at a logical high (1) by a 10kohm resistor. When the button is pressed, the pin's potential changes to logical low (0), and the microcontroller reads the “pressed” state. Through this button, the user can reset the system and initialize a new measurement.

Resistors R1-R8 maintain the input potential at logical low (0) in case the wire is interrupted. Otherwise, the measured potential could be affected by external disturbances or user handling, etc. The resistor values are not critical and were selected to consume as little current as possible from the microcontroller's pins, resulting in a current of 60uA for an 82k resistor.

Results



Conclusions

In conclusion, this project has helped us gain a better understanding of how to design, implement, and develop the necessary software for a device. We have learned valuable insights into working with microcontrollers, utilizing various input and output pins, interfacing with LCD displays, and performing cable continuity testing. Through this project, we have enhanced our skills in hardware design, programming, and troubleshooting. It has been a valuable learning experience that has equipped us with practical knowledge for future projects in device development.

Download

soft.zip

Bibliography

https://en.wikipedia.org/wiki/Category_5_cable

https://www.cnet.com/culture/how-to-make-your-own-ethernet-cable/ Datasheet ATMEGA16

Export to PDF